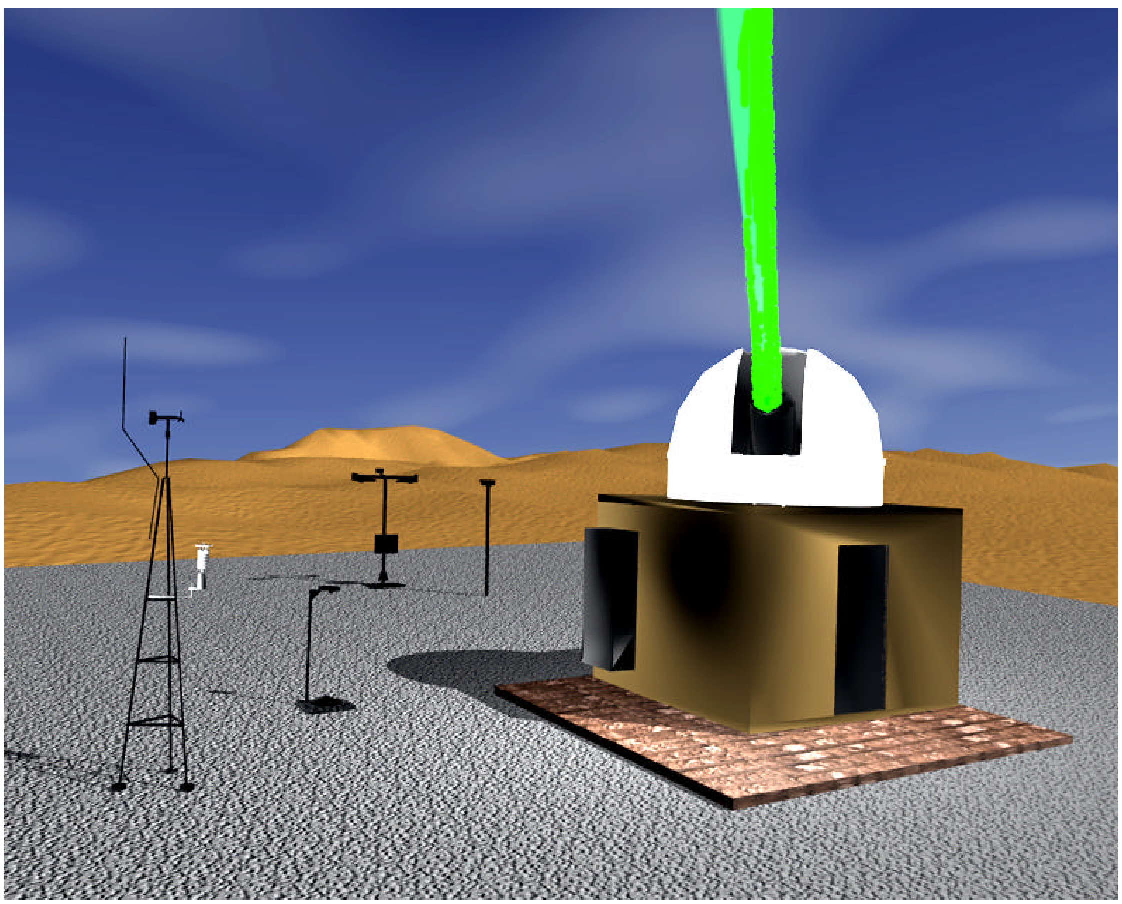
SLR2000

Software Design Document

November 13, 1998

NASA/GSFC



SLR2000
Software Design Document
November 13, 1998

Jan McGarry
NASA/GSFC

Brion Conklin
Anthony Mann
Mahtab Sadeghighassami
Mike Perry
AlliedSignal Technical Services Corporation

Jack Cheek
Tony Mallama
Nick Ton
Raytheon/STX

Randy Ricklefs
University of Texas at Austin

Acknowledgment

This work was supported under the Mission to Planet Earth office and the NASA Satellite Laser Ranging Project.

The software team would like to thank the following people for their support and inputs:

John Degnan, Tom Zagwodzki, Phil Dabney
*NASA/GSF*C

Win Decker, Bud Donovan, Dick Chabot, Mike Selden ,Jim Fitzgerald, Don Patterson
AlliedSignal Technical Services Corporation

Randy Pensabene , Joel Edelman
Orbital Sciences Corporation

Charlie Zelman
Aerotech

Paul Titterton, Hal Sweeney
Electro-Optics Organization

Disclaimer

This document has not been reviewed by anyone other than the authors, who are solely responsible for its content. All schedules in this document are subject to immediate and substantial change. The software design will change as the hardware design is finalized ;-)

TABLE OF CONTENTS

1) Introduction / Background

SLR2000 is the next generation of NASA's Satellite Laser Ranging Stations. These systems are expected to be deployed in the first few years of the 21st century. Unlike existing NASA SLR stations which require operators to determine system viability, avoid direct contact of the laser beam with aircraft and ground personnel, select the objects to track and interactively acquire and track, the SLR2000 system will be totally automated and completely eye-safe. To make the system eye-safe but ensure similar numbers of returns per second from satellites as from current system, the system has been designed to operate at a 2kHz fire rate (200 times faster than the existing norm). While the system itself will be much different than existing SLR stations, the final data products will be produced onsite as in current NASA SLR systems: Normal Point Data and MERIT II full rate. Normal Point data will be sent back to a external facility for distribution to the scientific community hourly as NASA SLR systems do now.

These stations will be placed so that they have full access to both Internet and phone lines (voice and modem communication). Each SLR2000 system will regularly communicate via Internet with an External Facility (called Home) which will send the system prediction data for the week, indicate the priority of satellites for tracking, monitor the system health and performance, and send maintenance technicians periodically and/or when the system needs repair. Normal Point data and MERIT II will be sent Home and/or to another archival and distribution facility.

SLR2000 will require extensive software development to replace the functions of the operator, gather data from and control the ranging system at a 2kHz rate, and manipulate and record large amounts of data from the ranging system, the telescope mount, cameras, health and safety monitors, weather instrumentation, and timing. To replace the operator and operate at 2kHz will require the development of many algorithms including:

a) Distinguishing the satellite signal from the background and detector noise

b) Determining if the system is healthy and if the outside conditions warrant opening the dome and operating

c) Knowing if the sky is clear where the satellite track occurs

d) Assessing the performance of the system, and alerting the home facility of the need for repair

e) Keeping track of each fire in flight and associating it with its corresponding return (due to the 2kHz fire rate, many fires will be in the air at the same time, and the software must keep track of the data)

f) Changing the laser fire rate to avoid collisions of fires with incoming returns

g) Scheduling the objects to track based on satellite priorities (given by Home) and the station mask (objects blocking the system view **--** such as a mountain **--** and time periods when the system isn't allowed to operate)

h) Focusing the system

i) Modeling and correcting for the mount pointing errors

j) Using quadrant detector and ranging data to correct time, angle, and range biases

k) Calculating the Point Ahead angle to transmit to the satellite and the Point Behind to receive the reflection from the satellite, and using these angles to drive the transmit/receive correcting mirror.

The design of this software system, including a description of the algorithms, the formats, and the hardware interfaces, is presented in this document. This should be considered a guide for the development of the SLR2000 software, and not a User's Guide. Flowcharts, tables, figures, timing diagrams, and data flow diagrams have been included where relevant (and where they were known). At this time the hardware design has not been completed, and much of the software design is, by necessity, still in flux.

2) Requirements and verification

2.1)<u>Requirements</u>

The SLR2000 system is required to perform laser ranging to the current constellation of satellites that the existing NASA SLR stations do, and will be required to support future spacecraft missions. It must acquire enough satellite returns per second to accurately generate normal points, must be able to schedule satellite passes according to a predetermined priority, and must generate and transmit Normal Points from the ranging returns. All this must be done without any human interaction either onsite or remotely. Specifically, the system must be able to acquire and track all satellites below GPS during the day or night, and GPS during the night. The system must be able to acquire all low and medium altitude satellites (including LAGEOS) above 20 degrees elevation, and all high altitude spacecraft above 30 degrees. Nominal acquisition times should be less than a minute for low earth orbiting (LEO) satellites and a few minutes or less for high satellites such as GPS. SLR2000 should get 6 returns or more per second from LAGEOS during tracking mode at least 80% of the time during clear sky. For LEO spacecraft this number of returns per second should be 60. For high altitude spacecraft such as ETALON and GLONASS, the return number per second should be 3 for all acquisitions above 30 degrees.

Once in tracking mode, the system should be able to stay in tracking mode (i.e. not lose the satellite and have to reacquire) except for the following valid problems: (i) sun avoidance, (ii) mount mask **--** object / time avoidance, (iii) sky hazy or cloudy, or (iv) hardware problems. Once tracking mode has been established, loss of signal, except in the above noted cases, should occur less than 10% of the time.

Satellite tracking should follow the predetermine priority as established by the Home facility. In the NASA case, this will be the CSTG priority list, with potentially minor changes for special tests or experiments. A log of daily events must be kept so that objects tracked can be compared against the schedule of satellites available.

The system must protect itself from harm, and alert Home when problems do occur. This includes not opening the dome when the external conditions would do harm to the system, powering off racks when internal temperatures or voltages get out of limits or when commercial power drops out, and avoiding pointing the telescope directly into sunlight. Emergency e-mails and calls to beepers should be sent whenever power is about to be shut off, or whenever hardware or external conditions reach extremes. The software must also assess the system performance and send home for help when performance begins to degrade.

Normal point data must be generated for each valid track, and sent to a central archival / distribution facility within 1 hour after the pass as long as the Internet is properly functioning. Merit II data must be available for special tests.

The NASA SLR project’s requirement for SLR2000 can be found in the document “ Requirements Document for the SLR2000 Fully Automated Satellite laser Ranging System “ by John Bosworth.

2.2) Software Verification and Test Plans

Any large software development project requires a well thought out verification plan to ensure success. The SLR2000 project is even more in need of such a plan due to its totally automated nature. All code paths MUST be tested and all decision processes checked. The Test Plan must be an integral part of the SLR2000 software development process.

The Software Verification Plan (SVP) details the tests that must be performed to determine compliance with the Software Requirements Document. These tests are system tests and must be performed with the final hardware configuration after all subsystem testing has been completed.

The Software Test Plan (STP) details the tests that must be performed to verify that each major section of code has been properly written, and that all code paths are correct. These tests are not system tests and while some must be performed with the hardware interfaces available, many can be performed using simulations. In fact, many tests can only realistically be completed using simulators in place of the hardware (since many hardware extremes are difficult to make happen without damaging equipment). Our test plan consists of the following steps (where not all are applicable to every piece of code):

1) Use the existing FORTRAN SLR2000 SIMULATOR (S2KSIM) and incorporate new algorithms for testing. The C code of the new algorithms can be combined with the existing FORTRAN code. Test the algorithms offline using S2KSIM to simulate all hardware.

2) Next, move the code to the actual SLR2000 computers (POP and ICC) and test using hardware simulation routines. These simulation routines would be called in place of inputs/outputs to actual hardware, and would be transparent to the code being tested since data would go into shared memory just as if the hardware were being accessed.

3) Next, test routines with hardware simulators where applicable. An example of this would be an electronic (no optical pulses) ranging simulator that uses the range gate output to return a pulse in the range window.

4) Finally, the routines must be tested with the actual hardware in subsystem tests and in final system checkout.

See the Software Test Plan Document for a detailed listing of all tests.

3) Overview

3.1) Introduction

The SLR2000 Software is divided into four functional sections that correspond to the four major CPUs (see section 4 on Computer Architecture). They are: (1) the Interface and Control Computer (ICC) which resides on the ISA/PCI bus Pentium, (2) the Pseudo-Operator (POP) is a Pentium single board computer residing on the VME bus running LynxOS, (3) the Data Analysis (DAN)is a Pentium single board computer residing on the VME bus running LynxOS, and (4) the Remote Access Terminal (RAT) which runs on the laptop running LINUX.

The ICC performs input and output to hardware with ISA or PCI interface cards. The software tasking is driven by 2kHz interrupts from the clock via the National Instruments PC-TIO-10 card. No data processing is performed on the ICC; all data is passed to and from POP via shared memory. This software runs under DOS.

POP controls most of the VME hardware interface, makes most of the operator decisions (thus its name), and passes data to and from DAN via files and shared memory. The software tasking is driven by a 2kHz interrupt from the clock.

DCS , the dome control system, is a micro controller based software package used to control the SLR2000 shelter dome and shutter. POP, using an RS232 interface, communicates with DCS at 1 Hertz. POP instructs DCS about the desired operational status and position angle of the shutter. DCS informs POP of the real operational status and position angle of the shutter. If requested, DCS will download to POP diagnostic information on DCS equipment. DCS can, in an emergency, close the shutter to protect the telescope. A detailed description of the communication between POP and DCS can be found in the document SLR2000 Dome Control Interface in the Appendix

DAN is responsible for generating the final data product (Normal Points) and for communicating with the external world. DAN gets predictions and priorities from Home via the Internet, and sends Normal Point data to the central distribution facility hourly. DAN also generates database information of tracking, weather and other data and makes this information available on a World-Wide-Web server. Maintenance of the system (software updates and hardware diagnostics testing) is via a user account on DAN. There is no direct logon into POP or ICC. A BVME Watchdog Timer card is used to reset the VME bus (and thus both CPUs) if DAN becomes inactive and fails to reset the Watchdog Timer's counter. DAN controls all system power, and is ultimately responsible for system shutdown, should that be necessary

The RAT software will reside on the laptop and will allow the operator to run software tests, perform diagnostic hardware tests, display data graphically on the monitor, and statistically analyze data. This laptop will be connected via the Internet to the SLR2000 system via DAN.

3.2)System drawing

Refer to figure 3.2

3.3) Data flow drawing

Refer to figure 3.3

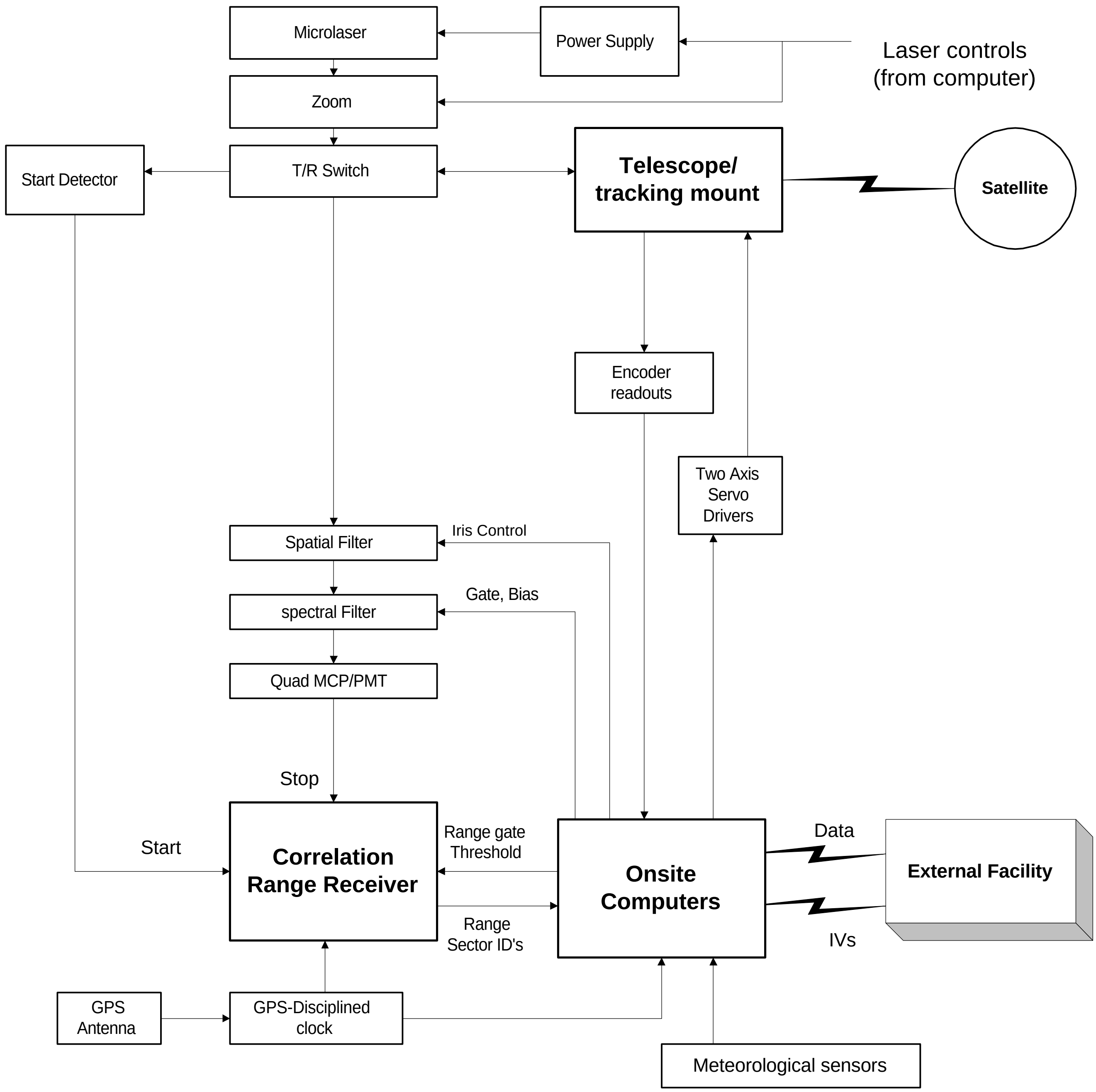


Figure 3.2

SLR2000 System Diagram

Originally from J. Degnan 9/15/97

SLR2000 DATA FLOW

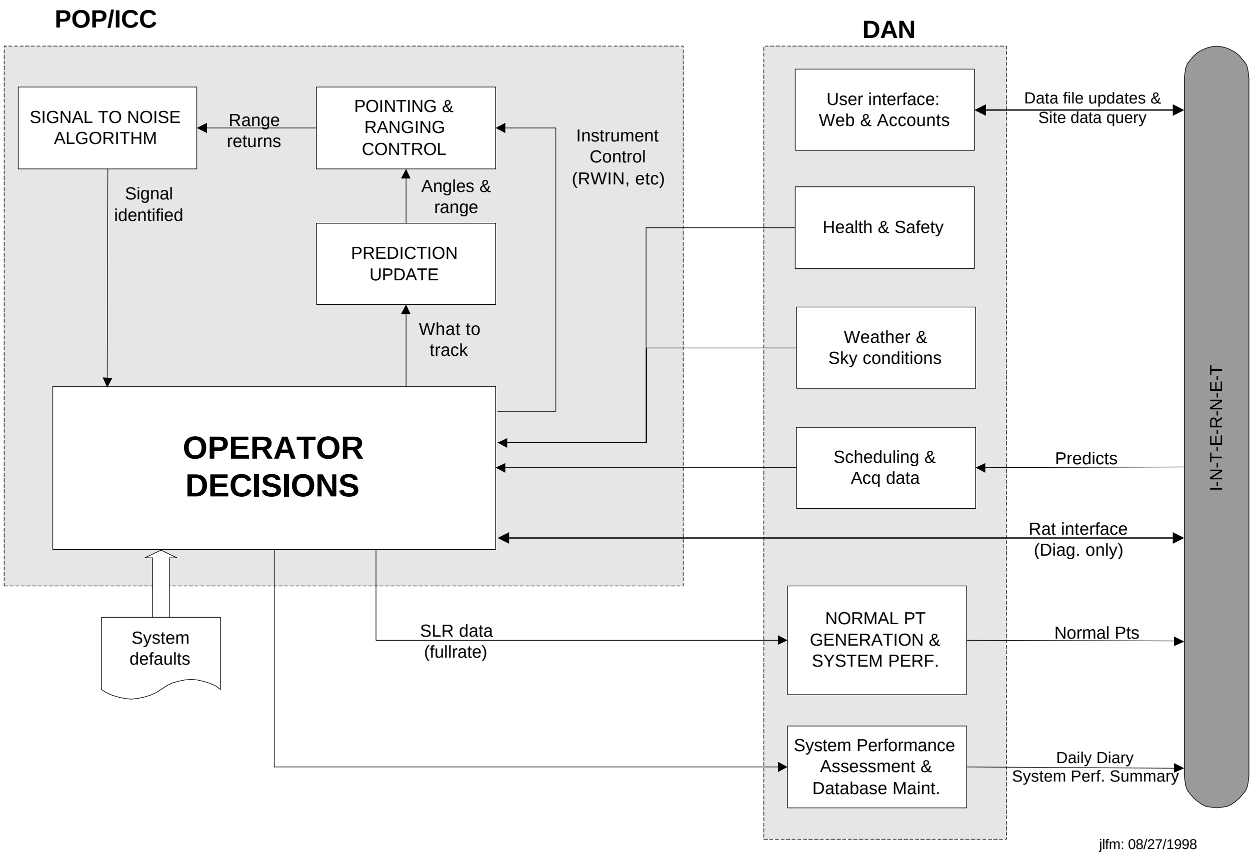
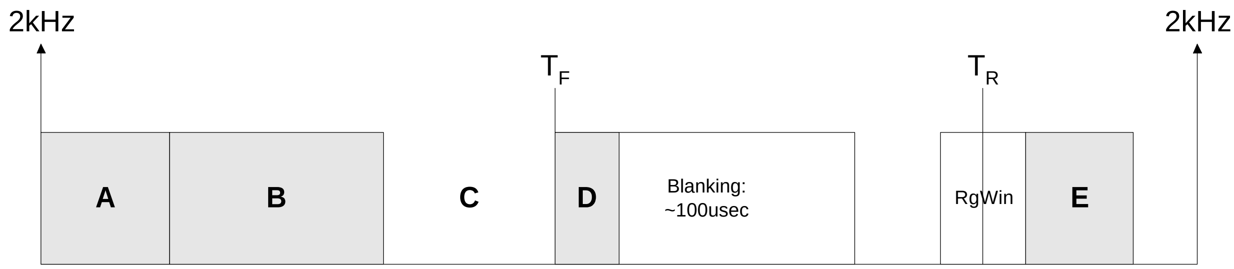


Figure 3.3

3.4) Timing diagram

3.4.1) ICC TIMING

Because of the 2kHz fire rate great care must be taken to ensure that all required functions get performed in the fastest possible manner. To accomplish the required tasks on the ICC in the 500 microseconds allotted, initiation of all ICC I/O must be done by interrupts. The following drawing illustrates this concept.



The 2kHz arrows mark the beginning and end of the 500 microsecond timing interval. The lines labeled T_F and T_R mark the Time of laser Fire and Time of range Return, respectively. Shaded areas are software tasks, and unshaded boxes represent hardware or other functions. Interrupts are generated at the 2kHz on time, the time of laser fire, and at the close of the range window. The software tasks are:

A: Clock functions (2kHz counter, 1pps reset, timing error checks)
Read all past intervals delays (Δt_F, Δt_G, etc)
Set buffer available flag in shared memory and reset pointer to other buffer for storage

B: Started via flag set in "A"
Mount input and output

C: Camera image exposure initiation (can't happen when laser is firing)

D: Change PRF by setting voltage (if necessary)

E: Read all range returns
Output range delay / range window settings for next interval

Note that the laser fire and range window may occur anywhere between the 2kHz markers; for instance the laser fire has as much probability of occurring after the range window as before. The range window, however, must never fall within the laser blanking time (which is around 100 microseconds in duration after the laser fire). To prevent this from happening, the software must change the laser PRF to move the laser fire time. There are a few problem areas that are unavoidable. Since the laser PRF should not be changed by a large amount (so the laser power isn't affected) the software cannot jump the fire time to the other side of the range window. To get the laser fire on the other side of the range window will require sliding the laser fire into the adjoining 2kHz interval creating two fires in the same interval. This should not be a problem since the software will only set up a range window for one fire in a given interval and thus the second fire and its corresponding return will be ignored. The real problem is that the range window will occasionally slide through the 2kHz on time. Depending on how the hardware is designed this could result in range returns getting missed. As long as the software prevents the range window from sitting in this state for more than a few consecutive intervals, this should not be a big concern, however.

There will need to be three distinct modes of ICC ranging: startup, operations, and shutdown. In startup, the laser is fired but no returns are yet expected. Just prior to the first interval where a return is expected, the range delay and window must be output in task A. During operations the software tasks are performed as shown above. In shutdown, the laser stops firing but the system must continue to generate range gates to receive the potential returns from the last few fires.

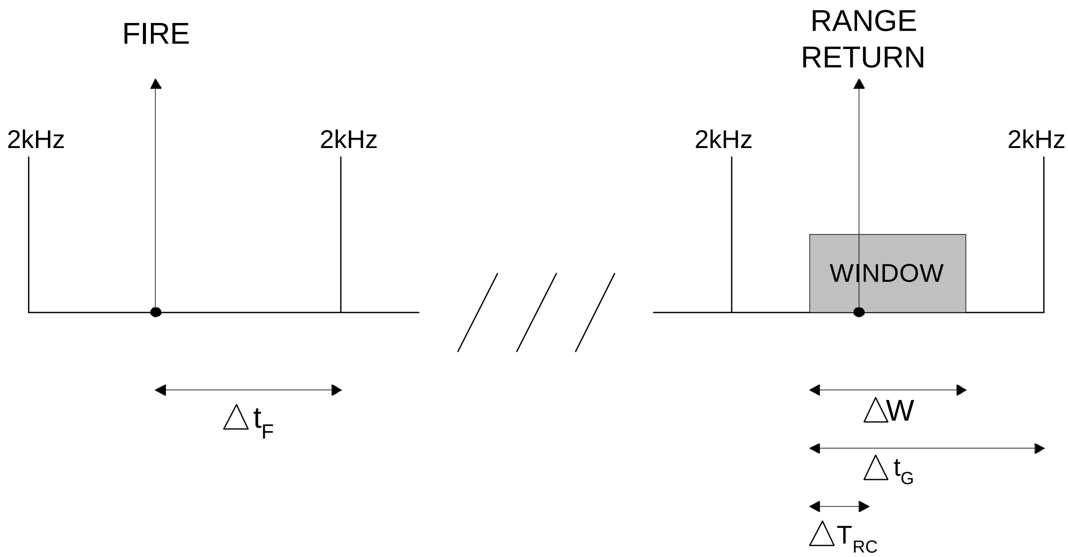
To keep the range data synchronized, the ICC must perform all ranging I/O, and data must be transferred between POP and the ICC at a 2kHz rate. This is because POP needs to know the laser fire time in order to accurately determine the corresponding return's delay into its interval. Calibration will have to be handled separately, however, since its return will occur within tens of microseconds from the laser fire. It would be advantageous to have a mode where the hardware opens the range window with the laser fire and an ND filter is used to attenuate the backscatter.

3.4.2) POP TIMING

POP will not perform any ranging I/O but will be required to compute the predicted ranges and range delays and output these to the ICC at 2kHz. It constructs the time of fire and return range times from the ICC data also at 2kHz. See the notes in section 3.4.3 on SLR2000 Data Timing and section 9.6 on Circular Buffer Algorithm.

While POP will not be required to do ranging I/O at 2kHz, the timing of shared memory transfer between POP and ICC is critical. POP will fill one ranging shared memory buffer during the 2kHz interval while the ICC is reading data from the other buffer. At the 2kHz on-time POP will immediately switch the shared memory buffer pointer to point to the buffer its been working on and turn on the buffer available flag. POP will then wait to see the ICC buffer available flag go on prior to picking up this ICC data for insertion in the circular buffer.

3.4.3) SLR2000 DATA TIMING



<u>Round trip Range of the Event is:</u> $\boldsymbol{\triangle T = M \cdot \triangle\tau + \triangle t_F - \triangle t_G + \triangle T_{RC}}$

<u>Time of the Event is:</u> $\boldsymbol{T = (N+1) \cdot \triangle\tau + \triangle T_{RC} - \triangle t_G}$

where:
τ = 500 μsec (2kHz)
t_F = time from FIRE to next 2kHz
t_G=time from range window opening to next 2kHz

And these times are computed f rom hardware readings:

$t_F = K_F \cdot\ t +\ t_f$
$t_G = K_G \cdot\ t +\ t_g$

where K_F,K_G are 10MHz counts read by hardware,
t_f, t_g are delays read by hardware, and t = 100nsec (10MHz).

T_{RC} = time from start of window to RANGE RETURN
N = number of 2kHz intervals from 1pps to the RANGE RETURN (software counts)
M = number of 2kHz intervals from FIRE to RANGE RETURN (software counts)

$$M = (INTEGER)\left[\frac{R_p - \Delta t_F - \Delta W/2}{\Delta\tau}\right] + 1$$

R_p is the predicted round trip range, and
W = RANGE WINDOW width.

4) Computer Architecture / Operating Systems / Language

4.1) Computer Architecture

VME was chosen for the main SLR2000 computer system due to its fast bus speed (40Mbytes/sec), its reputation for rugged dependability, and its long history of both laboratory and field use. Due to certain equipment selection, however, other backplanes were also required for the SLR2000 computer. The mount, star camera, and ranging electronics all come with specialized interface cards that only work in the ISA backplane. The PCI backplane was added due to its fast bus speed (130Mbytes/sec) for shared memory transfers to the VME. Unfortunately at the time of design, PCI was relatively new and many interfaces were not yet available for this backplane. Figure 4.1 shows the computer architecture including the interface cards and the shared memory connection between the ISA/PCI computer and the VME CPUs.

4.2) Operating Systems

LynxOS was chosen as the operating system for all the functions of SLR2000. LynxOS is a POSIX compliant realtime UNIX operating system which provides a good development and test environment and which has been used with great success on previous SLR systems. Lynx will reside on both the Pseudo-Operator CPU (VME) and Data Analysis CPU (VME).

LINUX is a non-realtime UNIX operating system which is very popular in the market today due to its price (free) and the availability of upgrades which keep the operating system current with new hardware. LINUX has been chosen for the Remote Access Terminal (Laptop).

The Interface and Control Computer (ICC) will run an in-house developed realtime tasking scheme over DOS. The DOME controller will use DOS as the operating system for the azimuth drive, and a simple in-house developed operating system for the shutter controller.

4.3) Languages

All software on SLR2000 will be written in C/C++ with the exception of:
(1) various drivers that may require assembly language
(2) the offline simulator (which is currently written in FORTRAN).

SLR2000 Computer System

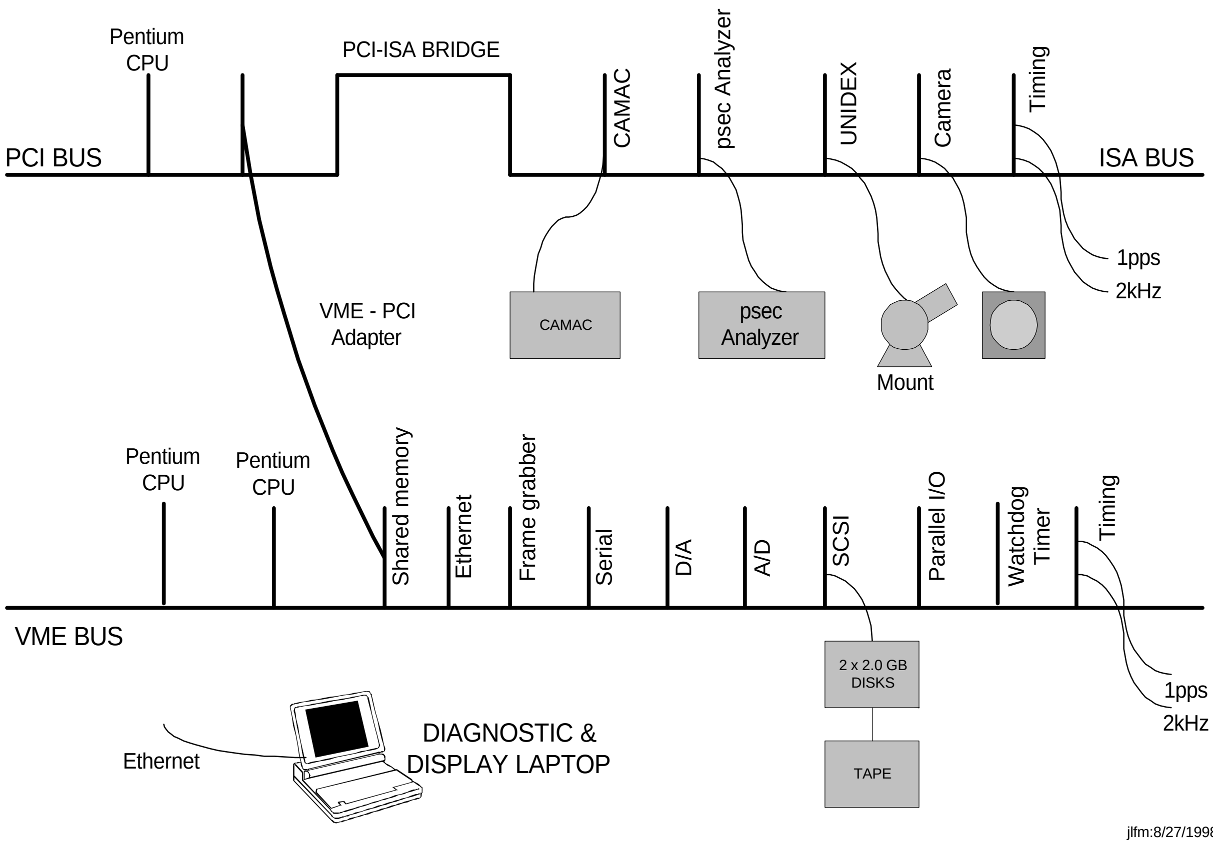


Figure 4.1

5) Off site interface

5.1) External Communication

5.1.1) Communication

The external facility is responsible for the maintenance of the SLR2000 site operations. To perform this duty communications from the external facility to the site will be supported in two forms, network and modem. The network is the primary means of communication. If the network is down, the site uses the modem to inform the external facility of that fact. A SuperTech or software engineer will be able to log onto DAN and perform various software and satellite maintenance functions as presented in the section below. Whenever there is a case that the site may be going down a message will be sent to the external facility via the network. When the UPS is in an emergency situation a pager will be called via a modem to inform the SuperTech of the situation.

5.1.2) Data

Information will be provided from both facilities. This information exchange will be routinely performed by DAN via ftp.

5.1.2.1) Data retrieved from the external facility:

- Daily satellite orbit predictions
- IERS.DUT
- Time bias file

5.1.2.2) Deposited Data onto the external facility:

- Normal point and engineering data provided on an hourly basis (to be archived and distributed)
- Daily Diary provided daily
- Weather data provided daily

5.1.3) Repair

The external facility is responsible for providing engineers familiar with the maintenance of the sites under its control. The persons called SuperTechs will be knowledgeable of both the hardware and software functions. The facility will maintain a spare parts set so that the SuperTech will be able to repair the site when necessary. The super tech will be able to perform diagnostics remotely or on-site via the RAT computer. The external facility may also be responsible for surveying the calibration piers and the site.

5.2) User interface

Each of the user interfaces on DAN (either accessed by logging into an account or through the WWW) provide users with functions to perform specific tasks.

5.2.1) Accounts

5.2.1.1) Maintenance

The Maintenance account provides the user the ability to check and diagnose certain aspects of the system. Functions are as follows:

OS Interface (shell)

Change Passwords

Software Patches
Control Overrides of subsystem
- Shutdown System
- Dome Close
- Mount Positioning

Tests Subsystems and instruments
Login Information
Disk Space Information
Backup
Retrieval

5.2.1.2) External Facility

The External Facility account provides the user the ability to change input parameters.
Functions are the following:
Set Parameters
Scheduling
- Mask
- Block Out Time File
- Schedule

Processor
- Satellites.dat
- Targets.dat
- Station Information Database

Retrieve Predictions

5.2.1.3) SuperTech

The SuperTech account gives the user the ability to perform diagnostics on the whole system.
Functions are as follows:
Includes all other previously defined functions
Logging
- Save Camera Data
- Save Health & Safety Data
- Save Pop Decision

View Data for Analysis
- Camera
- Health & Safety
- Pop Decision

Test Functionality
- Simulations
- Hardware
- Software
- Diagnostics

5.2.2) World Wide Web (WWW)

DAN will host the web server. The web pages will contain general information.
Scientific Area (password protected):
The Scientific Area provides the user the ability to review and gather information about calibration and satellite data. Functions are as follows:

Query Databases

Calibration
Satellite
Weather
System Performance
Retrieve Data

5.3) Emergency procedure

- Collect all of the latest housekeeping data into final message and send to the central facility.
- Power off the other racks
- DAN sends a message to POP to prepare for shut down
- DAN phones "home" with message and STATUS.
- DAN sends a command to POP to shut down
- DAN writes health and safety message
- DAN stays up until certain percentage of power is left on UPS.
- When power on UPS drops below level, DAN shuts itself off (time delay switch on power).
- UPS will turn on DAN and POP if power rises above a level.
- DAN restarts ICC when power is stable

6) Startup- Shutdown

6.1) Startup

- Power onto system brings up VME computer rack.
- DAN waits until UPS power is sufficiently high to bring up rest of system.
- Update system reboot file (keep track of reboots on POP/DAN)..
- Each CPU runs through diagnostics to check itself.
- After diagnostics, DAN begins handshake through shared memory with POP, and waits for a response. Reboot VME rack if no handshake after a period of time.
- After diagnostics, POP begins handshake through shared memory with DAN, and waits for a response. Reboot VME rack if no handshake after a period of time.
- After diagnostics, POP begins handshake to ICC and waits for response. (No response after a period of time is system error -- recycle power to ICC).
- After its own diagnostics, ICC begins looking for POP handshaking, and then responds.
- The Dome Controller (DCS) runs through its power up sequence, and then waits to see data from POP over the serial line. Shutter remains closed until POP commands it open.
- After all system conditions have been determined, DAN writes message to External Facility indicating system is up with system condition.

6.2) Shutdown

- Shut down all instrument operations (disable laser, close dome, stow mount, etc) and verify shutdown.
- Collect all of the latest housekeeping data into final message and send to external facility.
- Write final message to home (via Internet and modem) indicating going down.
- Power off all other racks (but VME).
- If condition is “BLACK” then send delayed command to UPS to shut off VME rack.

7) Hardware interface

7.1) Ranging and laser

OUTPUTS	CPU	#bytes	Interface type
Window (PMT)	POP/ICC	2 bytes	CAMAC
Delay (PMT)	POP/ICC	2 bytes	CAMAC
Bias (PMT)	POP/ICC	2 bytes	D/A
Window (psec Analyzer)	ICC	2 bytes?	ISA card
Delay (psec Analyzer)	ICC	2 bytes?	ISA card

INPUTS	CPU	#bits/bytes	Interface type
Lecroy counters	ICC	2 x 2bytes	CAMAC
Phillips Scientific TDC	ICC	2 x 2bytes	CAMAC
Delay (psec Analyzer)	ICC	4 x 4bytes	ISA card
TAG	ICC	4 bytes	ISA card

7.2) Telescope/Mount/Dome

Inputs/Outputs	CPU	#bytes	Interface type
Mount encoder	ICC	2 x 24bits	ISA card
Mount drive(pos,vel,acl)	ICC	2x(3x16bits)	ISA card (1-100Hz)(UNIDEX)
Other mount I/O?	ICC		
Dome Communication input	POP	7 bytes	Serial 1Hz
Cable wrap switches	ICC	2 x 2 bits	Digital
Mount home position	ICC	2 x 1 bit	Digital
Mount limit switch read-in	ICC	2 x 2 bits	Digital
Dome Communication output	POP	4-22 bytes	Serial1 Hz

7.3) Optical head

Inputs/Outputs	CPU	#bytes	Interface
Laser PRF voltage read	POP/ICC	2 bytes	A/D
Laser PRF voltage set	POP/ICC	2 bytes	D/A
Transmit power read-in	POP/ICC	2 bytes?	A/D?
Laser Enable/Disable	POP/ICC	1 bit	Digital
Focus setting read-in	POP	2 bytes?	???
Focus control (set)	POP	2 bytes?	???
Divergence zoom?	POP	TBD	???
Iris control? Iris state read-in?	POP	1 byte? 1 byte?	D/A? A/D?
Daylight filter set Daylight filter state	POP	1 bit 1 bit	Digital Digital
Transmit mirror (Pt Ahead)	POP	1 byte	D/A?

7.4) Timing

Inputs / Outputs	CPU	# bytes	Interface
1pps:level(200msec high;800msec low ;on-time)	POP&ICC	TBD	ISA&VME
2kHz: interrupt (10usec high;490 low;10usec after 1pps)	POP&ICC	TBD	ISA&VME
Date/time (GPS Date/Time : year set at boot from file)	POP&ICC	TBD	ISA&VME
Watchdog timer (output counter every 1 seconds),Timeout 10seconds	DAN	1 byte	VME
Other timers if necessary and possible from existing timer card	POP&ICC	TBD	ISA&VME

7.5) Cameras

Inputs/Outputs	CPU	#bytes	Interface type
Star camera image	ICC	242x242x1	ISA card
Star camera control	ICC	------	ISA card
Sky camera	DAN	120x120x2	Serial
Sky camera control	DAN	1 byte	Serial

7.6) Weather

Inputs/Outputs	CPU	#bytes	Interface type
Temperature	DAN	~10 bytes	Paroscientific,RS232,0.1Hz
Barometric Pressure	DAN	~10 bytes	Paroscientific,RS232,0.1Hz
Humidity	DAN	~10 bytes	Paroscientific,RS232,0.1Hz
Precip/Visibility	DAN	~50 bytes	Viasala:RS485:0.1Hz
Wind	DAN	12X2 bits	2 A/D

7.7) Health & Safety

Input /out puts	CPU	#bytes	Interface type
Temperature monitors	DAN	2 Bytes	Serial (VMIC 6016)
Voltage monitors	DAN	6 Bytes	Serial (VMIC 6016)
Frequency monitors	DAN	3 Bytes	Serial (VMIC 6016)
Current monitors	DAN	6 Bytes	Serial (VMIC 6016)
Vibration Motion switches	DAN	2 Bytes	Digital (VMIC 2511)
Smoke detectors	DAN	2 Bytes	Digital (VMIC 2511)
Water Sensor	DAN	2 Bytes	Serial (VMIC 6016)
Humidity monitors	DAN	2 Bytes	Serial (VMIC 6016)
Security cameras(panner and control)	DAN	1 Byte	Digital (VMIC 2511)
Security cameras	DAN	TBD	Raw Video (VMIC 6016)
Interlock switches	DAN	3 Bytes	Digital (VMIC 2511)
Rack's UPS	DAN	2 Bytes	Serial (VMIC 6016)
Control Monitors	DAN	6 Bytes	Digital (VMIC 2511)
Power supply monitors	DAN	2 Bytes	Serial (VMIC 6016)

7.8) Communication

-Internet.
- Phone/modem: backup comm, system reset, emergency beeper call.

8) Major functions

8.1) Interface Control Computer (ICC)

8.1.1) Range gate output

Runs at 2KHz and outputs the window for the psec Analyzer and possibly also the PMT window (if not done in hardware). The predictions come from POP already converted to the 500 usec time period. This routine picks up the right one and outputs it to the hardware (no computation involved). If the range flag is set to "NO" then the routine immediately exits without out putting to the hardware.

Inputs: 2kHz counter into second (from timing routine)
2kHz counter (from shared memory)
Range flag (from shared memory)
Range window (from shared memory)
Range predictions (from shared memory)

Outputs: (all to hardware)

8.1.2) Range receive

Reads the psec Analyzer delay and TAG. This information is then placed in shared memory for POP to handle. Runs at 2kHz. This routine will also read the EG & G 10MHz delays and the Phillips Scientific TDC. If the Range Flag is set to "NO" then the routine exits without reading any hardware or putting anything in shared memory.

Inputs: Range parts from hardware
2kHz counter from timing routine
2kHz counter from shared memory
Range flag from shared memory

Outputs: (all to shared memory)
Return range parts with counter into second

8.1.3) Mount command & position read-in

Reads mount position and status. Picks up mount commands from shared memory. Computes mount drive output values and outputs drive commands to mount. Runs at a maximum rate of 100Hz. If the Mount Flag is set to "NO" then no attempt is made to input from or output to the mount.

Inputs: Mount flag from shared memory
Mount commands from shared memory
2kHz counter to sync mount output (shared memory)
2kHz counter from timing routine
Data from mount interface card

Outputs: Buffer of mount positions with counter into second

8.1.4) Star camera read

Reads the star camera image and puts it into shared memory. If the Star Flag is set to "NO" this routine exits without reading the camera or putting anything into shared memory.

Inputs:	Star Flag from shared memory Star image from hardware 2kHz counter into second from timing routines
Outputs:	Star image with 2kHz counter into shared memory Camera status / error flag into shared memory Flag indicating star image available

8.1.5) Star camera control

Changes the settings on the camera (such as exposure time). This must be synchronized to the star camera read routine.

Inputs:	Star camera control info from shared memory
Outputs:	Camera status / error flag into shared memory

8.1.6) Laser enable/disable

This routine simply enables or disables the laser from firing. It is only called when the laser enable / disable flag changes in shared memory.

Inputs:	Laser enable / disable flag from shared memory
Outputs:	To hardware

8.1.7) Laser PRF control and voltage read-in

If the laser is enabled, then this routine will read in the voltage that controls the laser PRF. If the desired voltage as indicated in shared memory is different from that read-in, then this routine will output the new voltage setting to the hardware.

Inputs:	Laser enable / disable flag from shared memory Laser PRF voltage from shared memory
Outputs:	Laser PRF voltage as read from hardware Flag indicating whether a HW output was done

8.1.8) Timing routines (includes timing interrupt routine)

The timing routines keep track of the number of 2kHz intervals into the second and trigger software tasks. There is no actual read-in of the date/time on the ICC side. The timing interrupt routines get interrupts at 2kHz and a level for the 1pps signal. This allows the routine to determine which 2kHz interrupt is the start of the second. A counter that indicates number of seconds elapsed since the timing interrupt routines were started is also kept and placed in shared memory. An error condition occurs if the start of the 1pps occurs and the 2kHz counter doesn't read 1999, or if the 2kHz counter goes over 1999.

Inputs: 2kHz interrupts and 1pps level

Outputs: 2kHz counter into second
Seconds since start of tasks
Timing error flag

8.1.9) Range simulation routine

Simulates the range parts by using the range gate data. Different percentages of returns versus fires, noise percentage and error in return data can be simulated using information in the Test Parameters File on the ICC.

Inputs: Range simulate flag from shared memory
Range test # from shared memory
Range gate data from shared memory
2kHz counter from timing routines
2kHz counter from shared memory
Elapsed seconds from timing routines
Test parameters from file

Outputs: (all to shared memory)
Return range parts each with counter into second

8.1.10) Mount simulation routine

Simulates the mount positions by using the commands. The mount is assumed to have a limited acceleration and velocity, a wobble and a jitter. The amount of wobble, jitter, etc are contained in the Test Parameters File on the ICC.

Inputs: Mount simulate flag from shared memory
Mount commands from shared memory
2kHz counter to sync mount output (shared memory)
2kHz counter from timing routine
Test parameters from file

Outputs: Buffer of mount positions with counter into second

8.1.11) Star camera simulation routine

Simulates the star camera image. Gets the image parameters (such as brightness & image diameter) from Test Parameters File. The location (AZ/EL) of the image is sent from POP via shared memory. The routine must then compute where to place the star image in the camera FOV.

Inputs: Star simulate flag from shared memory
Location of star image (AZ/EL) from shared memory
2kHz counter into second from timing routines
Test parameters from file

Outputs: Star image with 2kHz counter into shared memory
Camera status / error flag into shared memory
Flag indicating star image available

8.1.12) Range diagnostics (TBD)

8.1.13) Mount diagnostics (TBD)

8.1.14) CPU diagnostic tests (TBD)

8.1.15) Shared memory diagnostic tests (TBD)

8.1.16) Timing diagnostic tests (TBD)

8.2) Pseudo Operator (POP)

8.2.1) Main control: OPERATOR (10Hz)

Operator provides the overall system control by doing the following:
- Gets analysis data from files, when flags indicate
- Reads control flags from shared memory
- Runs real time schedule
- Runs ACQ/TRK
- Runs simulations or diagnostics, if required
- Acts on Health & Safety and other alarms from DAN
- Keeps track of Sun location

This routine is always running, see section 10(POP flowcharts) ,the operator flowchart: 10.3a and b

Inputs:	Commands from DAN via shared memory Health & Safety system mode and data from shared memory Dome request flag from ACQ/TRK Files via NFS mount, as required Time/Date from shared memory
Outputs:	Operational system flag (Operations, Diagnostics, Simulations) to shared memory Dome open/close flag to shared memory Sun location to shared memory

8.2.2) Real time schedule

Initialization opens the schedule file and locates the satellite available at the present time(performed once per minute). Checks the new target flag from ACQ/TRK and if it is on, locates the next priority satellite. Decrements the time left count and locates the next target when the counter is zero. See section 10 (POP flowcharts) figure 10.5

Input:	Schedule (file) Present date and time (shared memory) Change target flag (shared memory)
Output:	(All via shared memory) Target id Start time in seconds of day Duration in seconds of day Targets starting azimuth and elevation Targets prediction file name Change target flag

8.2.3) ACQ/TRK (10Hz) and related threads

ACQ/TRK runs in conjunction with the Operator thread (section 8.2.1). Using the Realtime-Schedule subroutine, it determines what to track and starts the appropriate threads: SAT_ACQ/TRK, STAR_ACQ/TRK, STAR_ASSES anc CAL_ACQ/TRK for satellites, star calibrations, star assessments and ground calibrations, respectively. These threads in turn control running of the signal detection, data recording, quadrant detection, star camera solution, mount model update, and ground calibration solutions. See the descriptions under the ACQ/TRK algorithm section and flowchart in 'POP flowcharts' ;section 10 (figure 10.4).

Inputs:

All:

- Realtime schedule (type of event)
- Mount data
  - Sky clarity

Satellite and/or Ground Calibrations:

- SIC
- starting location
- tracking data
- calibration target
- ground visibility

Star Calibrations or Star Assessments:

- star table
- star camera image solution
- focus information

Outputs:

All:

- Object being tracked to shared memory

Satellite tracking and Ground calibrations:

- tracking mode (ACQ vs TRK) to shared memory
- update of circular buffer (see circular buffer section)
- angular search biases to shared memory
- quadrant detector to shared memory
- time and range biases to shared memory
- range window setting to shared memroy

Star Calibration and Star Assessment:

- star camera image biases to shared memory
- mount model update
- focus data

8.2.4) Dome control

POP controls the dome system through communications with DOME Control System (DCS). The communication protocol will be RS-232, in raw mode. POP sends command packets that DCS will parse and use to direct the dome to the desired location and state. This command packet resides in POP shared memory and will be assembled and sent to DCS through a low priority thread once a second. For more details see the appendix.

Inputs: Binary semaphore from timing thread
Shared memory command packets (See the Appendix on DCS Command Packets)

Outputs: Assembled command packet
Parsed return data packet from DCS

8.2.5) Performance assessment

Will determine the performance of the system by looking at ground calibrations, star calibrations, and ability to acquire and track. The output of this assessment will determine if either a new star calibration will be performed (unscheduled) or if a message should be sent home requesting maintenance. See the Performance Assessment Algorithm section for more details. This routine runs as a cron job hourly.

Inputs: Last starcal RMS from Starcal Summary file
Last ground calibration RMS and system delay from file
Acquisition percentage from file

Outputs: Flag in shared memory indicating performance OK or problems.
Problem indicator in shared memory.

8.2.6) Range data construction (2kHz)

Picks up the ICC range information along with circular buffer predicted ranges and constructs the fire and return times, places this information in the circular range buffer for use by signal detection routine. This routine is spawned by SAT_ACQ/TRK. See the Range Prediction, Construction and Circular Buffer Algorithm section for more details.

Inputs: Range pieces from ICC shared memory

Outputs: Range return and fire information in Circular Buffer (Shared Memory)

8.2.7) Range prediction (10 Hz & 2kHz)

Computes the range for some number of fire intervals ahead, places this into the circular range buffer, controls the laser PRF, separates the data into the parts required for gating, and places it into shared memory for ICC and shared memory for other threads. This routine is spawned by SAT_ACQ/TRK. See the Range Prediction, Construction and Circular Buffer Algorithm section for more details.

Inputs: Date/time via shared memory
Satellite or target pointing angle from prediction subroutine
Existing circular buffer with pointers in shared memory

Latest time bias, range bias and range window info in Shared Memory
Current laser PRF in shared memory

Outputs: Predicted range into circular buffer
Laser PRF setting
Range gate (delay & window) for expected return range in ICC shared memory

8.2.8) Angle prediction (=<100Hz)

Computes the azimuth and elevation commands for the next mount drive interval. Adds all biases from Q.D., angular scan, and time-bias. Also computes expected angle for point-behind. Corrects the angles to drive around the telescope mask and around the sun. Spawned by SAT_ACQ/TRK.

Inputs: Date/time from shared memory
Satellite or target vector from file or RA/DEC from shared memory
Last range predict for point-ahead / behind (if satellite)
Time bias in shared memory
Angle biases in shared memory

Outputs: Azimuth and elevation position, velocity and acceleration in shared memory

8.2.9) Camera solution thread (5Hz)

Reads the camera image buffer and computes the centroid of the star image. This centroid location in pixels is differenced with the known center of the receiver field of view and converted to an Az, El offset by scaling, image rotating, and correction for elevation.

DX = Xcentroid - Xcenter
DY = Ycentroid - Ycenter
DXP = DX*cos(theta) + DY*sin(theta)
DYP = -DY*sin(theta) + DX*cos(theta)
where theta is the image rotation angle
AZoffset = DXP*AZscale
Eloffset = DYP*ELscale

Inputs: Camera image buffer in shared memory
FOV center and scale factors in shared memory

Outputs: AZoffset and Eloffset in shared memory

8.2.10) Data buffering thread and control of ICC shared memory (2kHz)

Controls the flow of the ICC shared memory buffer. The ICC shared memory buffer is double-buffered and flags direct the threads to which one is currently being filled for the output data, and another to indicate which input buffer is available to get data from. This routine runs at the start of each 2kHz time interval and switches the POP flag over to point to the buffer just filled by POP. It also sets the POP data available flag on. It then waits until it sees the ICC buffer available data flag go on, and then picks up the corresponding pointer for the ICC buffer.

Inputs: Flag indicating which ICC buffer is to be read (ICC shared memory)
Flag indicating ICC buffer data available (ICC shared memory)

Flag indicating which POP buffer is being filled (shared memory)

Outputs: Flag indicating which POP buffer is to be read (ICC shared memory)
Flag indicating which POP buffer is being filled (shared memory)
Flag indicating POP data available (ICC shared memory)

8.2.11) Data recording (2kHz)

Operational data is written to a file for DAN to retrieve approximately once per hour. A flag is set in POP-> DAN shared memory to indicate when the file is available. The format is an extension of the New Controller format (See section 12; Data Formats, Controller Range Data Formats). During testing all returns can be logged, but under normal operating conditions will record only those intervals that indicate a signal return. This thread will be tightly coupled with the SAT_ACQ/TRK, Ground_Cal and Circular buffer threads.

Inputs: control flags indicating which records are to be written
data from the circular buffer
data from shared memory

Outputs: file records and file open/close
flag to indicate to DAN that a file is available
file name of the file so DAN can retrieve it

8.2.12) Star calibration thread

When a star calibration is scheduled, the ACK/TRK thread turns off the associated laser and ranging functions and spawns the star calibration thread. The star calibration has its own prediction thread, camera solution thread, and data recording thread. When the star calibration completes successfully, it spawns the mount model update thread to generate the mount model coefficient solution.

Inputs: Date/time from shared memory
Mount angles from shared memory
Camera images from shared memory
Sky clarity from shared memory
Weather parameters from shared memory

Outputs: Star biases versus angle and time (star cal data) in file

8.2.13) Mount model update

Generates the mount model coefficients from the star calibration data. The star cal data is obtained from file, and the mount model coefficients and summary are also written to files. A flag is updated in shared memory to indicate to DAN that the star calibration has completed and
the files are ready to be picked up (if desired).

Inputs: Star calibration data file

Outputs: Updated mount model coefficients
Star calibration summary file
Flag in shared memory indicating star calibration complete

8.2.14) Star assessment

The system will point to a star between satellites tracked. To be successfully used, the star must be along the path from one satellite to another (meaning within 10 degrees and in the correct cable-wrap path), and must be a high enough magnitude. The star will be used to (i) assess the pointing offset, (ii) focus the system, and (iii) provide yet another indicator of sky conditions. It may also be used to update the mount model using a Kalman Filter update.

Inputs: Mount angular information in shared memory
Star camera image in shared memory
Receiver FOV center and scale factors in shared memory
System focus information in shared memory
Sky clarity in shared memory
Star magnitude in shared memory
Filter / iris information in shared memory

Outputs: Az/El offset of star from center of FOV
Updated focus information in shared memory
Az/EL star diameter in shared memory
Updated mount model (maybe)
Sky clarity update (maybe) for this AZ/EL location

8.2.15) Ground calibration solution

After any ground calibration, POP will calculate the system delay mean and standard deviation and percentage of returns versus expected. Outputs from this routine will be used as inputs to the Performance Assessment Algorithm. This routine will be spawned by the ACQ/TRK routine.

Inputs: Range returns in shared memory
Predicted ranges in shared memory
Number of valid returns and number of fires in shared memory

Outputs: System delay mean and standard deviation appended to file
Percentage of valid returns vs fires appended to file

8.2.16) Signal detection (signal to noise)

Determine which of the returns are signal and which are noise. Flag the signal returns in the circular buffer and return with a flag indicating SUCCESS (we found the signal) or FAILURE (we didn't). This routine is called once per frame by the SAT_ACQ/TRK routine. For a more detailed description see the Signal detection Algorithm section.

Inputs: Circular buffer in shared memory

Outputs: Updated flags in circular buffer indicating Signal or Noise

8.2.17) Quadrant detector angular correction computation (Frame)

Count the number of signal returns in each quadrant and use this information to determine an angular offset for correcting the mount pointing. This routine is called by the SAT_ACQ/TRK routine.
Call the signal count in the four quadrants: Q1,Q2,Q3,Q4. Compute:

X=(Q1+Q2-Q3-Q4) / Qtot
Y=(Q1+Q4-Q2-Q3) / Qtot

Where Qtot = Q1+Q2+Q3+Q4

After scaling and image rotation (X,Y) is the correction that must be applied to the telescope (Az, El) to point directly to the target

Inputs: Circular buffer in shared memory

Outputs: Az,El biases computed from the Q.D. in shared memory

8.2.18)Timing(2kHz interrupt)

Keeps track of 2kHz interval count and seconds of day. At 0.1Hz rate it spawns a process to read the date/time from GPS, and then compare with its internal date/time. Reads the 1pps level on every interrupt and resyncs the 2kHz counter with the 1pps every second. Error flags are set if either the 1pps occurrence doesn't match with 2kHz counter or the seconds of day counter doesn't match with the GPS date/time.

Inputs: Hardware interrupts
GPS date/time

Outputs: Date to shared memory (day of year and year)
Seconds of day to shared memory
Number of 2kHz intervals since start of second (0 - 1999)

8.2.19) Simulation of range data

Operates with ICC to simulate the range data. Runs at 2kHz. When this routine runs, the ICC is told to simulate ranging functions. This routine provides simulation parameters in the diagnostic shared memory area for ICC

Inputs: Range parameters for simulation in shared memory (from Ratsnest or file)
POP shared memory buffer flags

Outputs: Range parameters for simulation to ICC shard memory
Shared memory buffer flags

8.2.20) Simulation of angular data

Similar to the range simulation, this routine interacts with ICC to simulate the mount angles . Runs at 100Hz. When this routine runs, the ICC is told to simulate mount functions. This routine provides simulation parameters in diagnostic shared memory for ICC

Inputs: Mount parameters for simulation in shared memory (from Ratsnest or file)
POP shared memory buffer flags

Outputs: Angle parameters for simulation to ICC shared memory
Shared memory buffer flags

8.2.21) Simulation of camera image

Interacts with ICC to simulate the star camera image by providing simulation parameters in ICC diagnostic shared memory. When this routine runs, the ICC star camera simulation flag is set.

Inputs: Star camera simulation parameters in shared memory from Ratsnest or file
(Including star diameter, brightest, location, etc.)

Outputs: Star camera simulation parameter in ICC shared memory

8.2.22) Simulation of starcal & solution

Generates a simulated star summary and mount model update.

Inputs: Simulation parameters via shared memory from Ratsnest or file
(such as starcal RMS, number of stars, etc)

Outputs: Star summary file
Mount model update file

8.2.23) Hardware simulator interface

Subroutine to control the hardware simulator. This routine is TBD since the hardware simulator has not yet been designed.

8.3) Data Analysis Computer (DAN)

8.3.1) Processing Overseer

-Watch dog timer
- Checks shared memory for flag set by POP indicating data is ready to be processed.
- Sets a flag to inform POP that DAN is still alive.
- Moves files off of POP and onto DAN via NFS mount.
- Splits data into individual files of calibration and satellite data.
- Runs Calibration processor and process all calibration data sets first.
- Runs Satellite process and processes all satellite data sets.
- Runs Normal Point processor
- Health and safety
- Weather system
- Watchdog timer

Inputs: Data ready flag (from shared memory)
Data filename (from shared memory)

Outputs: Data finished flag (to shared memory)
Thread active flag (to shared memory)
Data filename

8.3.2) Satellite Processor

- Gets system delay information from Calibration database
- Reads controller file
- Loads information from databases
- Computes Refraction Correction
- Creates Merit II Record
- Calculates mean and rms for the corrected range, temperature, humidity, and pressure.
- Creates Satellite Capture Record

Inputs: Data filename (input parameter)
Site Geodetics (file)
Satellite parameters (file)
Calibration Database

Outputs: Merit II satellite data
Satellites Capture Record

8.3.3) Calibration Processor

- Loads information from databases
- Reads calibration into array of structures
- Computes refraction correction
- Corrects the range
- Calculates the mean and rms for the corrected range, temperature, humidity, and pressure.
- Determines system delay
- Writes system delay to calibration database
- Creates calibration capture record

Input: Controller file (calibration data)
Target database
Geodetics.dat(file)

Output: Calibration capture record
System delay database record

8.3.4) Normal Point Processor

This process reads the merit 2 full rate and produces CSTG normal points.

- Loads information from data files
- Reads tracking information into structures
- Computes satellite residuals
- Computes corrected satellite orbit
- Computes an nth order polynomial
- Calculates the mean and rms for the pass.
- Determines each normal point bins corrected range and bin rms
- Writes CSTG Normal points

Input: Merit II data file (calibration data)
Satellite database
Geodetics (file)
Gravity model (file)
Output: CSTG normal points (file)
Normal point summary (file)

8.3.5) ACQ data generation (Cron job once per day)

- Retrieves tracking predictions from central location
- Generates the ivs file for POP
- Sets flag informing POP of the new predictions

Input: TIVs (file)
Satellite database
Geodetics (file)
Gravity model (file)

Output: IVS (file)
Data ready flag (to shared memory)

8.3.6) Schedule generation (Cron job once per day)

- Reads the schedule parameters file
- Determines the start and end times for all of the available satellites
- Schedules passes according to satellite and event priority
- Outputs the schedule
- Informs POP of the new schedule

Input: IVs (file)
Schedule parameters (file)
Geodetics (file)

Output: Schedule (file)
Data ready flag (to shared memory)

8.3.7) Performance assessment(Cron job once per day)

- Reads the performance parameters file
- Compares the previous day's star calibration data, ground calibration and satellite data to performance specification Parameters
- Flags data outside of performance specifications Output for Daily Diary

Input: Performance specifications (file)
Star calibration database (file)
Calibration database (file)
Satellite summary (file)

Output: Daily Diary problem (file)

8.3.8) Data distribution

Cron performed once per hour to send out the normal point data and archive the data on site
- Sends CSTG normal point data to central location
- Archives data

8.3.9) Daily Diary(Cron once per day)

- Reads in the past 24 hours of site activity
- Reads Site H & S problem file
- Reads Performance assessment problem file
- Forms the diary message
- Distributes the diary message

Input: Performance assessment problem (file)
Star calibration database (file)
Calibration database (file)
Satellite summary (file)
H & S database (file)

Output: Daily Diary (file)

8.3.10) Sky camera read

When the Sky flag goes from "NO" to "YES", a command is sent to the sky camera for an image, and the RS232 input interrupt is enabled. At the completion of the image download (which may take seconds), a flag will be turned on in shared memory to indicate that the sky camera data is available. This routine periodically checks for the completion of the RS232 download.

Inputs: Sky flag from shared memory
Sky image from hardware

Outputs: Sky image
Camera status / error flag into shared memory
Flag indicating sky image available

8.3.11) Maintenance

Process to be started upon logon to a maintenance account that will allow for site, satellite, scheduling and other parameters to be changed.

8.3.12) Diagnosis and Testing

A process to be started upon logon. This may start a real time thread between the analysis computer and POP to allow for tracking system commands to be generated on the analysis computer.

8.3.13) Sky camera simulation routine

Simulates the sky camera image. Gets the sky parameters (such as percentage of clouds in each of 8 or 16 sky sections) from Test Parameters File. The routine must then compute a value for each pixel in the sky map according to the parameters.

Inputs: Star simulate flag from shared memory
Location of star image (AZ/EL) from shared memory
2KHz counter into second from timing routines
Test parameters from file

Outputs: Star image with 2KHz counter into shared memory
Camera status / error flag into shared memory
Flag indicating star image available

8.3.14) Simulation of Health & Safety

POP makes individual system statuses (such as domestat, laserstat, etc.) available in shared memory for Health and Safety to read. Health and safety then simulates a system status for all the hardware readings (including all the sensors and MET system) using a uniformly distributed random number generator. The mean and standard deviation of the random number probability density is read from a file hence easily changeable. The POP generated statuses and Health & Safety simulated statuses are analyzed and the worst condition is picked as the overall system status. A time tagged error message is written every time any of the statuses change as well as the reason. No message is written for the continuous green status.

Input: System statuses from shared memory
Parameters from file for random number probability distribution

Output: Randomly generated system status for hardware readings
Overall system status to shared memory

8.3.15) Health & Safety (1Hz)

Gathers information about health and safety of the system and analyzes them to produce a system status. It then makes this system status available to POP via shared memory. Gathering the information is done in different ways such as reading the sensors directly, obtaining subsystem status from POP via shared memory and obtaining meteorological information from the algorithm that is running independently on DAN. Health and safety algorithm sets the operationally safe bounds for each system. It then compiles the information from each subsystem and determines if it is safe to operate. There are six possible system status which are presented with different colors

SENSORS	***STATUS***
NO Data	***White***
All sensors and subsystems are within bounds, it is safe to operate	***Green***
Some subsystems are approaching the bounds but it is ok to operate	***Yellow***
Some subsystems have crossed bounds, stop operation and run diagnostics	***Orange***
Several crucial subsystems are malfunctioning; turn off power for selected racks.	***Red***
Catastrophic situation or security intrusion occurred shut down everything; Turn off all power	***Black***

Inputs: Hardware: Serial and Digital I/O from following sensors: Temperatures, humidity, smoke detectors, voltages, current, frequency, interlocks, Security Cameras

Software: The following status from shared memory: DomeStat, Met System, LaserStat, MountStat, Ranging ElecStat, StarCam Stat.

Outputs: System Mode to shared memory

8.3.16) Sky Clarity Assessment

Determines viewing conditions. Gets the sky camera data from DAN. This subroutine will compute a map of the sky clarity using two bits per Az/El location (0=no data,1=clear, 2=hazy, 3=cloudy). The resolution of the Az/El is expected to be around 1 degree square. This routine is expected to run at around 0.01 Hz.

Inputs: Sky camera temperature map from hardware
Scale factors for temperature conversion in shared memory

Outputs: Map of sky clarity versus Az/El location in shared memory

8.3.17) Watchdog Timer

Loads the watchdog timer counter at periodic intervals to prevent the watchdog timer from resetting the system.

Inputs: None

Outputs: Value sent to register in hardware to reset counter.

8.3.18) Weather Data Input

Provide required weather information. Send commands from DAN to the weather instrument and collect serial and analog data from them. The serial instruments are the Paroscientific which provides Temperature, pressure and humidity, and the Vaisala which gives precipitation and visibility information. The analog instruments are the Belfort/Young which measures wind speed and direction, and the DRD12 rain detector.

Inputs: Serial and analog data from weather instruments.

Scale factors for analog data in shared memory.

Outputs: Temperature, pressure, humidity, precipitation, visibility, wind speed, and wind direction in shared memory.

8.3.19) Weather Data Simulation

Simulate required weather information. The method for simulation is currently to set constant weather parameters from an ASCII editable file. Later versions of the software will include parameters in shared memory for stochastic processes which will dynamically model the data.

8.3.20) Sky camera simulation

The sky camera is simulated by reading in a previously acquired thermograph file and using a simulated ambient temperature to determine clear, hazy, or cloudy conditions. By changing the ambient the simulated sky conditions can be controlled. Later versions of the software may contain the ability to randomly generate the sky camera output.

8.4) Remote Access Terminal (RAT)

Even though SLR2000 is designed to be an unmanned ranging station, during the initial system development and debugging as well as during regular maintenance and upgrades of these stations, some type of human interface beyond a shell prompt will be required. To fulfil this need the Remote Access Terminal is being developed. The RAT accesses the SLR-2000 system via an information and command server (Ratsnest) on the SLR2000 Data Analysis System (DAN) across a Berkley socket interface and via Network File System(NFS). The user interface (Ratgui) is located on a laptop. It is mainly used for diagnostics and systems testing purposes, and will not be used in normal operation It receives and displays data to accomplish the following tasks:

8.4.1) Examine current ranging system status and faults

Server: Ratsnest provides access to realtime information. It runs as a background information and command server whenever DAN is operating. Ratsnest acts on commands sent by Ratgui and passes DAN and POP shared memory information to Ratgui via TCP sockets

Client: Ratgui receives and displays real time status information such as weather information, telescope position and system status alarms, from Ratsnest.

8.4.2) Display current and recent data of many types

Server: Ratsnest provides access to data files that are to be displayed by Ratgui via Network File System. Displays include:(see section 15 for examples)

➢ Debug file
➢Ranging log file
➢ Post fit O-C
➢ StarCal analysis result (such as sky map, mount model map and sky error map)
➢ MET system historical data vs time
➢ Camera display window(including sky camera, star camera and security camera)

Client: Ratgui provides graphical user interface to SLR2000.

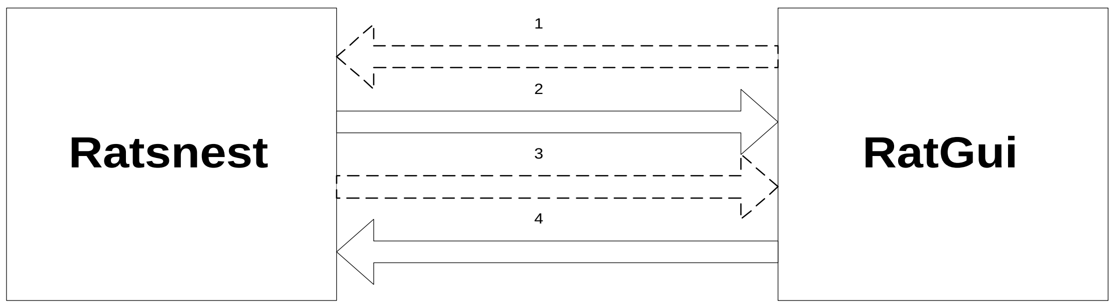
8.4.3) Override system operating parameters and decisions

Server: Ratsnest receives override flags and values via a TCP socket from RAT and applies them as necessary, usually by making changes to shared memory.

Client: Ratsnest provides pop up windows and similar interfaces to set and send the override flags and values to Ratsnest via a socket.

8.4.4) DAN/RAT communications

Ratsnest and Ratgui socket communications follows the scheme shown in figure 8.4.4



Host Data Analysis Computer (DAN)

Multi-threaded Socket Server

Accesses POP/DAN Shared Memory

Can support multiple instances of RatGui

0) DAN listens for socket request(s)

1) RAT requests command socket

2) Command socket established

3) DAN reuests data/ reply socket

4) Data/reply socket established

Host: Remote Acess Terminal (RAT)

Dual-threaded Socket Client

Accesses Shared memory via Sockets

Accesses Files via NFS

DAN/RAT Communications

Figure 8.4.4

9) Algorithms

9.1) Health & Safety

The Health and Safety algorithm gathers information regarding the health and safety of the system, analyzes this data and determines the system mode which it makes available to the pseudo operator once a second. Some of this information is gathered by reading various sensors and some is obtained from POP and DAN via shared memory. The health and safety algorithm has been divided to three major sections: Facility, Site Security, and the system status obtained from shared memory (See figure 9.1a).

Under facility, main power is monitored by sensing the current, voltage and frequency on the load side for each phase once a second. If irregularities are observed then current is checked before the breakers; and current and voltage is checked for the racks(to find out the root of the problem). The power is controlled on the line side. This enables health and safety to shut down the power in catastrophic situations. Current and voltage on the racks are monitored every minute unless the main power control shows problems. There is an intelligent power controller on each rack that can shut down each rack individually in case of irregularities. The temperature and humidity are sensed every minute in the Dome and Electronics room while the smoke detector in each is monitored every second.

Site security consists of two subsections: the interlocks and security cameras. Four interlocks are monitored every second. These are: compound door, station door, access to roof, and hatchway to dome area. There are three cameras: camera number one is the compound camera which monitors the outside area with pan, tilt and control options. Camera number two, in dome area, is stationary with a wide angle lens and zoom option. Camera number three is in the electronics room with same options as the dome camera. A fourth camera is possible to observe the blind side of the compound area. All cameras take a snapshot of their default field of view every thirty minutes but if interlock sensors show irregularities or the possibility of intrusion then the camera is prompted to take a picture of the interlock at which the intrusion occurred .

Various system statuses are conveyed to the health and safety algorithm by POP and DAN through shared memory. Dome status provides the current information about the dome such as whether it is open or closed and why. The MET system consists of the Paroscientific (temperature , relative humidity, barometric pressure), the Young wind sensor(wind speed and direction), the Vaisala visibility monitor (horizontal visibility and precipitation). The MET system parameters are observed by health and safety every ten seconds. LaserStat, MountStat, Ranging ElecStat, StarCamStat are updated every second and placed in shared memory by POP

Although not all the data are updated every second, the overall system status to POP via shared memory is updated every second.

The health and safety criteria for each status color is illustrated in the tables 9.1b and 9.1c.(For each color definition see Health and safety function :section 8.3.15) Table 9.1b (weather parameters) is Goddard Geophysical and Astronomical Observatory(GGAO) specific. These are only educated guesses which will be adjusted as the software development and testing continues and they will be stored in a site specific file later.

SLR2000 HEALTH AND SAFETY

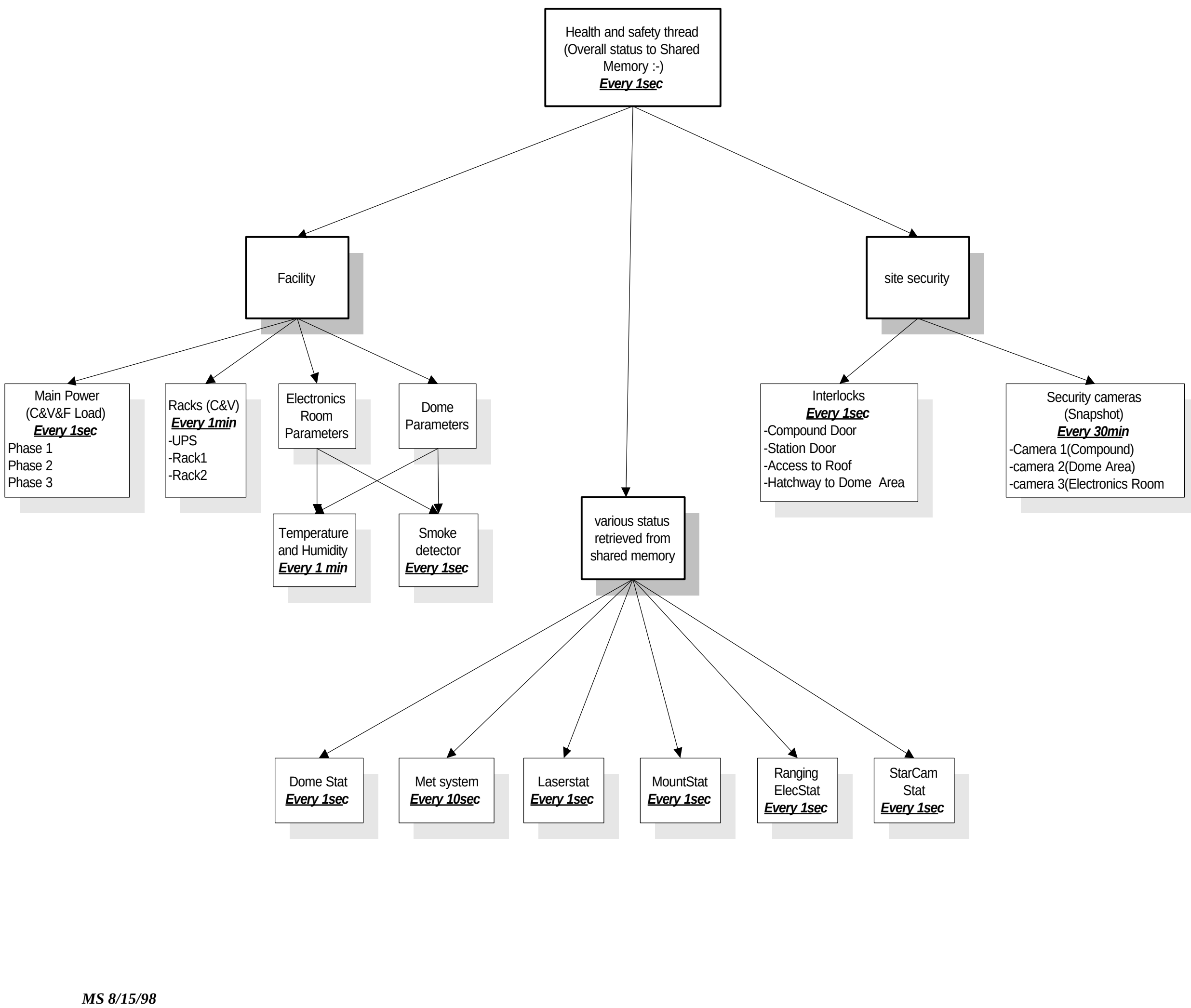


Figure 9.1a

Weather parameters

SENSORS\STATUS	Green	Yellow	Orange	Red	Black
Sustain wind speed (m/s) (averaged over 1 minute)	0 to 8.941	8.941<V<17.882	V>17.882	X	X
Outside Temperature(C)	10<T<35	-10<T<10 & 35<T<45	T<-10&T>45	X	X
VAISALA (Precipitation)	No	X	Yes	X	X
VAISALA (Visibility)km	vis>3km	1km<vis<3km	vis<1km	X	X
Outside Barometric Pressure (GGAO) (mbar)	P>985	920<P<985	P<920	X	X
Analog voltage from rain sensor *	4 (no rain)	X	X	X	X
Dew / frost point (Temperature/humidity)(C)	T> D+3	D< T < D+3	T=< D	X	X
* This sensor is only check to reconfirm the "no rain" status when the VAISALA shows "no rain"					
D -> Dew point					
T -> Temperature					
P -> Barometric pressure					
Vis -> visibility					
V -> wind speed					
m -> meters					
s -> seconds					

Figure 9.1 b

Hardware instruments

SENSORS\STATUS	Green	Yellow	Orange	Red	Black
Security Camera 1 (outdoor) Status	yes pic	no pic	NA	NA	X
Security Camera 2 (Electronics room) Status	yes pic	no pic	NA	NA	X
Security Camera 3 (Dome area) Status	yes pic	no pic	NA	NA	X
Main Power Phase 1 Fence side Current	TBD	TBD +5%	TBD +10%		X
Main Power Phase 2 Fence side Current	TBD	TBD+5%	TBD+10%		X
Main Power Phase 3 Fence side Current	TBD	TBD+5%	TBD+10%		X
Main Power Phase 1 Load side Current	TBD	TBD+5%	TBD+10%	0 Amp.	0 Amp
Main Power Phase 1 Load side Voltage	105-125Vac	104-126Vac	<104->126V		X
Main Power Phase 1 Load side Frequency	57-63Hz	56-64Hz	<56->64Hz		X
Main Power Phase 2 Load side Current	TBD	TBD+5%	TBD+10%		X
Main Power Phase 2 Load side Voltage	105-125Vac	104-126Vac	<104->126V		X
Main Power Phase 2 Load side Frequency	57-63Hz	56-64Hz	<56->64Hz		X
Main Power Phase 3 Load side Current	TBD	TBD+5%	TBD+10%	0 Amp.	0 Amp
Main Power Phase 3 Load side Voltage	105-125Vac	104-126Vac	<104->126V	X	X
Main Power Phase 3 Load side Frequency	57-63Hz	56-64Hz	<56->64Hz	X	X
Rack1 Intelligent power controller	Green	Yellow	Orange	Red	Black
Rack1 Current	TBD	TBD +5%	TBD +10%	0 Amp.	0 Amp
Rack2 Intelligent power controller	Green	Yellow	Orange	Red	Black
Rack2 Current	TBD	TBD +5%	TBD +10%	TBD	0 Amp
Roof door Interlock	Closed	NA	Open	NA	X
Building door interlock	Closed	NA	Open	NA	X
Front gate interlock	Closed	NA	Open	NA	X
Building door vibration motion switch	Closed	NA	Open	NA	X
Electronics Room Smoke detector	Closed	NA	Open	Open	
Electronics room Temperature	TBD	<60 or >80F	<32 or >100F		
Electronics room Humidity	non-condensing		condensing	condensing	condensing
Electronics room Water sensor	Closed	NA	Open	Open	X
Dome area Smoke detector	Closed	NA	Open	Open	X
Dome Temperature	Ambient	<TBD>ambient	<TBD>ambient		X
Dome Humidity	Ambient	<TBD>ambient	<TBD>ambient		X
Dome Water sensor	Closed	Closed	Open	Open	X
					X
Note: Many of the above conditions need to be logically anded or ored to determine system status					

Figure 9.1c

9.2) System Performance Assessment

This version of system performance assessment will reside on POP to help POP assess the need for star calibrations. This algorithm will use past ground calibration RMSes, past star calibration RMSes, and information gleaned from the ability to acquire satellites, to determine if a star cal should be performed, and/or if "home" should be notified of need for maintenance. In order for system performance to hold the following nominal task should occur:

1) Last starcal RMS < 5 arcsecond
2) Last calibration RMS < 2 centimeter
3) Percentage of acquisitions vs trials for LAGEOS and below in clear sky >= 80 percent
4) System delay change between the last two calibrations < 1 cm

9.3) Signal Detection Algorithm

The Correlation Range Receiver (CRR) algorithm distinguishes the satellite returns (Signal) from the background noise and detector dark counts (Noise) by using the knowledge that the noise follows Poisson statistics. The algorithm is most important and has the biggest challenge during acquisition when knowledge of the system and orbital biases are worse and the range window must be larger. A histogram is computed in O-C space from the returns in the range window. The actual satellite returns tend to fall in a single or a few bins, while the noise is fairly uniformly distributed. With our improved knowledge of the orbit (due to daily ephemeris updates) the slope of the satellite data in O-C space is relatively small, and the histogram bin size can be made to accommodate the spread of the data across this small slope.

There are two version of the CRR algorithm: the Single Frame version, and the N of M version. In the Single Frame version, data is collected for a period of time over which 10 satellite returns are expected (this period is called a frame). return ranges are differenced from the predicted (O-C) and placed in a histogram. At the end of the period, any bin with six or more counts is a candidate for being satellite returns. The bin with the most counts is declared the winner. If no bin has counts above 5 then it is assumed that no satellite returns are in the data. For the existing SLR2000 system parameters (40cm telescope, +- 4 arcsecond divergence, +- 5 arcsecond field of view of the receiver, 133 microJoule transmit energy, 1.2 nm bandpass filter, etc) this produces a 90% probability of success which is achievable in theory and simulations for Starlette at 20 degrees or above, LAGEOS for 22 degrees or above, and ETALON for 30 degrees or above (day or night). The probability of false alarms is negligible for Starlette and less than 10% for LAGEOS and ETALON.

Table 1 shows some examples of Frame times and expected noise levels for each of the above satellites in the Single Frame CRR case. In each of these examples the range window size is 0.5 microseconds and the histogram bin size is 0.5 nanoseconds, thus the histogram will have 1000 bins. Note that for LAGEOS and ETALON the noise counts per Frame are 625. Spreading 625 noise counts over 1000 histogram bins results in an expected noise count per bin per Frame of 0.625, making 6 or more noise counts in a single bin a very low probability event.

Table 1: SLR2000 daylight satellite acquisition

Satellite	Altitude	Acq. Elev.Signal	Expected photoelectrons	Frame Length	Noise cts per Frame
STARLETTE	800 km	20 deg	0.0090 pe	0.6 sec	30
LAGEOS	5900 km	20 deg	0.0004 pe	12.5 sec	625
ETALON	19000 km	30 deg	0.0004 pe	12.5 sec	625

The N of M case has promise for satellites whose predictions are not as well known. In this technique the algorithm looks at multiple frames instead of a single frame. N frames out of M must all have a histogram bin with 6 or more counts to achieve acquisition. Tests for the 2 out of 3 case demonstrate that this technique is better able to handle high noise counts, and may permit acquisition of a signal that is walking through adjacent bins as a result of prediction error. Both versions will be coded for use in the SLR2000 system.

Single Frame Version
Initial algorithm setup:
Calculate frame time from the expected signal level:

K_s is a constant made from many system parameters, τ_a is obtained from cloud information, τ_c is based on cloud information and satellite elevation, σ is the satellite lidar cross section and is obtained by table lookup, and R is the predicted range to the satellite. Also N_s is the number of expected signal returns required for a 90% probability of actually getting k, and k is the number of counts per bin required to establish signal. In our current case Ns = 10 and k=6. This may later be changed to allow k to be determined by the noise counts, and a lookup table (or calculation from Poisson statistics) will give the value of Ns required for 90% probability of success.
Make the frame time an even multiple of 0.1 seconds by rounding up to the nearest 0.1 seconds.

Per shot functions:
IF there is a return in the range window, THEN:
 Difference range with prediction
 Compute histogram bin from above difference
 Increment that bin's count
 Increment background counter

Per frame functions:
Determine if any histogram has a count greater than or equal to 6.
If none is greater than to equal to 6, THEN
 Set NO ACQUISITION flag.

If one or more bins has a counter greater than or equal to 6, THEN:
 Pick the bin with the greatest count and return its number.
 Set the ACQUIRED flag.
 Recalculate the frame time based on the formulae in the initial setup above.
 Zero histogram counts, and setup for next frame.

* N of M version *
Initial setup is the same as the Single Frame, except that the values and the tables will be different. The per shot calculations are also the same.

Per frame functions:

Determine if any histogram has a count greater than or equal to 6.
If one or more bins has a counter greater than or equal to 6, THEN:
 Pick the bin with the greatest count and save its number.
 Zero histogram counts, and setup for next frame.

IF there are N frames with valid bins, THEN
 Set ACQUIRED flag
 Return saved bin numbers for all valid Frames.
 Recalculate the frame time based on the formulae in the initial setup above.
 Zero any N of M counters and setup for next M-Frame event.

ELSE
    IF Frame=M, THEN
        Set NO ACQUISITION flag

Recalculate the frame time based on the formulae in the initial setup above.
Zero any N of M counters and setup for next M-Frame event.

9.4) Satellite search

Due to the narrow laser divergence coupled with orbital uncertainties and mount positioning errors, SLR2000 will often have to institute a search for the satellite in both angle and range. Once the pointing offsets are acquired in angle and range, this information is then converted (if possible) to time and range. The angular search pattern is a rectangular spiral in AZ and EL. The angular step size is 1/4 of the beam divergence. The actual azimuth and elevation steps corresponding to this are:

$azstep=\theta_{div}/\cos(el)/4$
$elstep=\theta_{div}/4$

where θ_{div} is the half angle divergence. Some limit must be made on cos(el) prior to division (such as limit el to 89 degrees). The range step should be 1/4 of the range window. A step is only taken when no returns have been seen for some given period of time, and the SAT_ACQ/TRK algorithm has determined that the sky is clear enough to get returns. The algorithm should take a range step followed by angular search. The range delay biases (range steps) should have the following sequence:

+rstep,-rstep,+2*rstep,-2*rstep, ..., +n*rstep,-n*rstep.

A nominal value of n=4 can be used, although this number may be increased or decreased depending on the accuracy of the predicts. The angular search should also have a limit on the length of the search. When this limit is reached, SAT_ACQ/TRK can decide to abandon this satellite or to restart the search pattern (perhaps with some other parameters changed). A nominal area for the angular search would be to cover +- 20 arcseconds in both axes. The parameter "lensearch" is related to the area searched and the laser divergence by the following formula:

Area searched = $S_{az} \cdot S_{el}$, where
$S_{az}=lensearch \cdot \theta_{div} / \cos(el) / 4$, and
$S_{el}=lensearch \cdot \theta_{div} / 4$.

Thus to cover 40 arcseconds per side the length of the search would have to run out to at least 40 steps (depending on where in elevation the search was being done).

See the flowchart 10.8a and 10.8b in section 10 (POP flowcharts) for the details of the algorithm.

9.5) Sky Clarity Determination

The SkyCam consists of a 'ThermaSnap' 10 micron infrared camera pointed down toward a convex aluminum coated mirror that reflects the whole sky. The camera is sent an 'I' command, over an RS-232 line, to record an image of the sky. It scans the focal plane, where the reflected sky is imaged, with its array of 120 silicon thermoelectric IR detectors. Then the 120x120 array of 16-bit thermal fluxes and a 200-byte header are downloaded from the camera into the computer over an RS-232 line.

The azimuth and zenith angle corresponding to each pixel will be computed based on the SkyCam's geometric calibration. Each temperature will be adjusted based on the radiometric calibration, which is a function of zenith angle. The ambient air temperature at the site, which has been acquired from another routine, is accessed, and the 'surface minus sky' temperature difference is computed for each pixel.

Pixels where the 'surface minus sky' temperature exceeds a certain TBD limit (probably about 20 degrees Celsius) will be marked as clear, those less than another TBD limit (probably about 10 degrees) will be marked cloudy, and those in between will be marked hazy.

9.6) Range Prediction, Construction, and Circular Buffer Algorithm

The Circular Buffer resides in POP's local memory and holds all of the ranging and angular information over a period of time equal to four GPS Frames. (See the Signal Detection Algorithm for a definition of Frame). This buffer is a tool to correctly associate the return ranges with their corresponding laser fires and to hold the data until the Signal Detection Algorithm can process the information and the Logging Thread can record the data to file.

The Circular Buffer is separated into a Ranging Buffer and an Angular Buffer since these two types of data are updated at different frequencies. Each entry in the Range Buffer corresponds to a 500 microsecond interval. A maximum of one fire and four returns can be recorded per entry. Each entry in the Angular Buffer corresponds to a command to the mount (which could occur as fast as 100Hz). The data in the Angular Buffer can be tied to the corresponding data in the Ranging Buffer by the time stamp associated with each entry. Pointers keep track of the current buffer entry (corresponding to current time). When the last entry location in either of the buffer's array has been filled in, the pointer wraps around to the top of the buffer and begins refilling these entries with new information (thus the name circular buffer). At the end of a Frame, the Circular Buffer places the indices of the start and end of the Frame into POP's local shared memory for access by the Signal Detection Algorithm, and ultimately for use by the Data Logging Thread. The Signal Detection Algorithm will then fill in the Range Buffer "N[i]" values to indicate whether the returns were determined to be signal or noise.

The format of the buffers is:

```
/* ---------------------------------------------------------------------------- */
/*  Circular Buffer Structures (Jan McGarry/NASA/GSFC/920.3)   */
/*  October 13, 1998                                          */
/* ----------------------------------------------------------------------------*/

//  Angular Buffer: All angles are in degrees
struct AngElem{
long   I2K;              /* # of 2KHz interval past the buffer start time */
long   tbias;            /* time bias (usec) */
double azenc;             /* azimuth position - measured */
double elenc;            /* elevation position - measured */
double azbias;           /* total azimuth bias */
double elbias;           /* total elevation bias */
double azcmd;            /* commanded azimuth (predicted + biases) */
double elcmd;            /* commanded elevation (predicted + biases) */
double azrate;           /* azimuth rate (deg/sec) - measured */
double elrate;           /* elevation rate (deg/sec) - measured */
double azqd;             /* azimuth bias from Quadrant Detector */
double elqd;             /* elevation bias from Quadrant Detector */
double azscan;           /* azimuth scan bias */
double elscan;           /* elevation scan bias */
double azmanual;         /* manual bias input from RAT */
double elmanual;         /* manual bias input from RAT */
```

```
double azmm;          /* mound model calculation */
double elmm;          /* mound model calculation */
double elref;         /* refraction correction calculation */
double azptahead;     /* transmit point ahead angle (difference from receive) */
double elptahead;     /* transmit point ahead angle (difference from receive) */
double azdome;        /* dome measured location */
};

struct AngCircBuf{
double AngT0SOY       /* Start of circular buffer in seconds of year */
struct AngElem[12000]
};

// Ranging Buffer: all integers are in picoseconds - all reals are in microseconds
// All data, except where specified, is associated with THIS interval's fire
struct RngElem{
long   I2K;           /* # of 2KHz interval past the buffer start time */
long   dtF;           /* delta fire time (for this interval's fire) in psec */
double Rp;            /* predicted range for this fire (usec) */
double Rpdot;         /* predicted range rate for this fire (usec) */
long   tblank;        /* blanking time for this interval (psec) */
long   MR;            /* number of intervals to this fire's expected return */
long   dtE;           /* expected time into THIS interval for return from previous fire (psec) */
long   MF;            /* # of interval back to fire for return(s) in this interval */
long   dtR[4];        /* measured range delta(s) from start of interval for this fire (psec) */
double R[4];          /* measured roundtrip range for this fire (usec)*/
char   Q[4];          /* tag indicating quadrant for each return (1,2,3,4, or 0) */
char   N[4];          /* flag indicating signal(=1), noise(=2), or no data(=0) */
double rbias;         /* range bias (usec) */
long   tbias;         /* time bias (usec) */
long   rwin;          /* range window width (psec) */
double dRref;         /* refraction correction (usec) */
};

struct RngCircBuf{
double RngT0SOY       /* Start of circular buffer in seconds of year */struct       RngElem[240000];
};
```

Each entry in the Ranging Buffer has 120 bytes. The current maximum Frame size (corresponding to GPS) is 30 seconds. Thus the Range Buffer will require:

120 bytes/entry · 2000 entries/second · 30 seconds/Frame · 4 Frames = 28.8 Mbytes

Each entry in the Angular Buffer has 168 bytes per entry. This corresponds to:

168 bytes/entry · 100 entries/second · 30 seconds/Frame · 4 Frames = 2.0 Mbytes

for a total of around 30 Mbytes required for the entire Circular Buffer.

During interval “i” the Circular Buffer Thread will perform the following tasks. In what follows $\Delta\tau$ represents the 500 microsecond interval between the 2kHz on-times.

A. Range construction for interval “i-k” and angle retrieval (if data available)

1. Get “dtF” from ICC data and store in circular buffer entry “i-k".

2. m = i-k - MF(i-k)

3. Get “dtR” from ICC data and store in circular buffer entry “m”.

4. Construct return range(s) and store in interval “m”:

$$R(m)=MR(m)\cdot\Delta\tau + dtR(m) - dtF(m)$$

5. If angular data is available from ICC, then place in correct Angular Buffer slot.

B. Range prediction for interval “i-k-1" and angle prediction (if time for mount output)

1. Calculate TF(i-k-1) = RngT0SOY + I2K(i-k-1)/2000 + dtF(i-k-1)

2. Calculate predicted roundtrip range:

$$Rp(i\text{-}k\text{-}1) = F\ (\ TF(i\text{-}k\text{-}1) + Rp(i\text{-}k\text{-}2)/2 + tbias\ ) + rbias$$

and store it with time and range bias into slot “i-k-1".

3. Compute number of intervals to the expected return:

$$MR(i\text{-}k\text{-}1) = INTEGER\ [\ (Rp(i\text{-}k\text{-}1) + dtF\ (i\text{-}k\text{-}1) - rwin/2) / \Delta\tau\ ]$$

4. Calculate the buffer entry number:
m=i-k-1 + MR(i-k-1), and store the window width there.

5. Compute the expected delay:
$dtE(m) = Rp(i\text{-}k\text{-}1) + dtF\ (i\text{-}k\text{-}1) - rwin(m)/2 - MR(i\text{-}k\text{-}1)\cdot\Delta\tau$

6. Store the number of intervals back to the fire, MF(m)=MR(i-k-1), into “m”.

7. Using same tbias and rbias as in the range prediction, compute mount angles and store into Angular Buffer.

C. Change laser PRF, if necessary

The nominal PRF setting is 2kHz. However, the PRF must be changed if the range return will fall within the blanking time (approximately 100 microseconds after the laser fire). Changing the PRF in advance of a problem can slide the returns and prevent them from falling within the blanking time. This is an ongoing process and must be continually monitored and corrected for during the course of a pass.

D. Transfer shared memory with ICC

The predicted data (range and angle) must be sent to the ICC every 2khz interval, and the range and angular measured data must be retrieved from the ICC at the same rate.

E. Zero out ahead in Circular Buffer

Since the Circular Buffer will be wrapping around back on itself, this thread must clear out the data for about 0.5 seconds ahead in the buffer to ensure there is no confusion.

F. Keep track of Frame time

At the end of a Frame the Circular Buffer Thread will post the start and end indices for the current Frame into shared memory. The POP OPERATOR will then call the Signal Detection Thread which will use these indices to determine the signal search area.

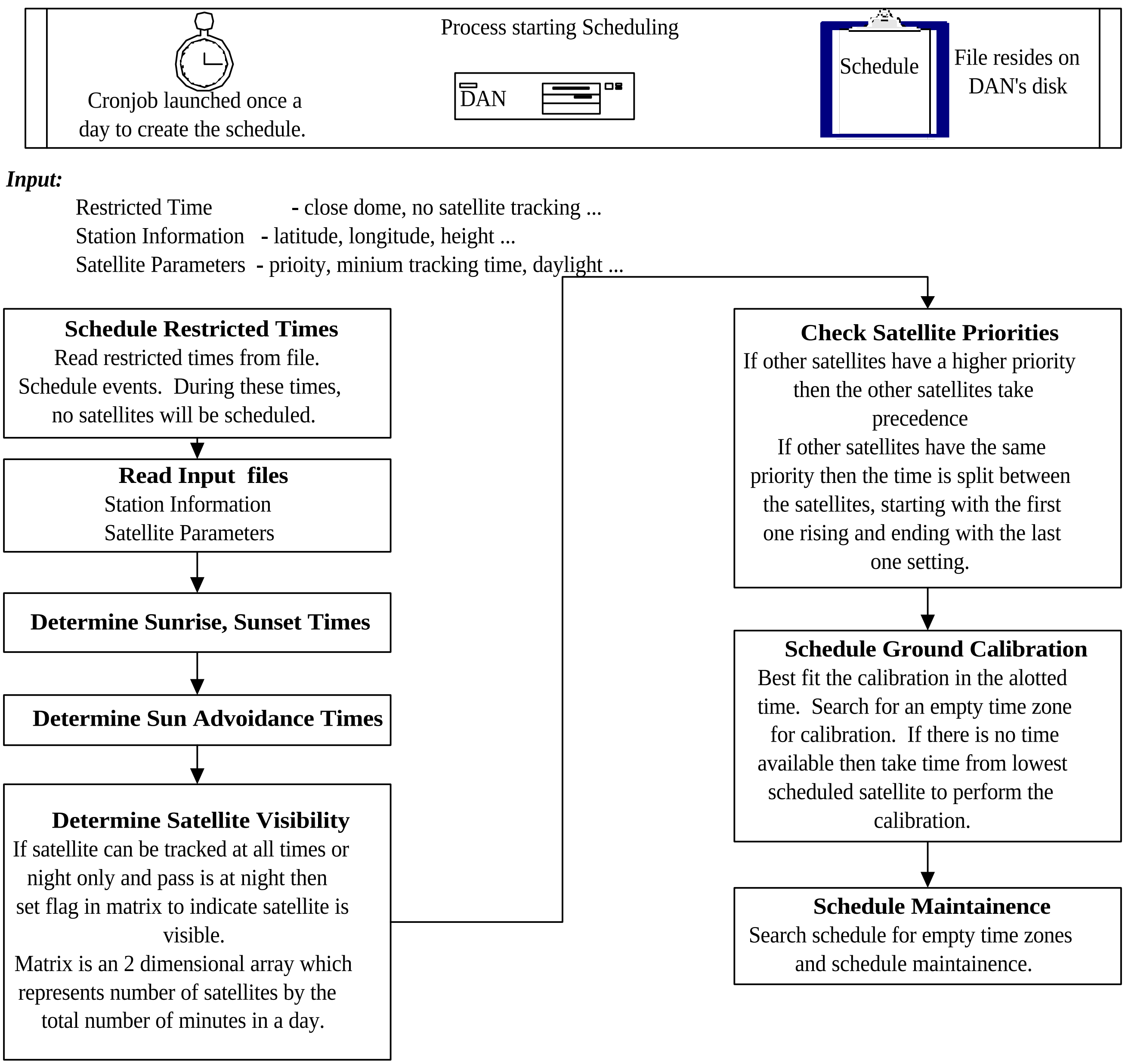
9.7) Acquisition, Tracking and Scheduling (ATS)

ATS controls tracking operations, deciding what to track, when to adjust biases, what to adjust them to, when an object has been acquired, and when an object should be rejected. ATS leaves the system health monitoring to the OPERATOR routine and concentrates on the objects to be tracked. ATS is also responsible for determining when the system performance warrants doing a non-scheduled star calibration. Figures in section 10 (POP flowcharts) show the ATS flow and the flow of the tasks spawned by ATS including SAT_ACQ/TRK, Star_Assessment, and STAR_ACQ/TRK. The CAL_ACQ/TRK is not shown since its flowchart is the same as SAT_ACQ/TRK's except that instead of looking at the sky clarity, CAL_ACQ/TRK looks at the ground visibility.

9.8) Non-Realtime Schedule

The satellite predictions for azimuth and elevation for each satellite will be done in one minute intervals. Starting at the highest priority satellite each pass is scheduled so that if a higher priority satellite is already scheduled that satellite is a secondary target not primary. Calibrations are then scheduled such that the higher priority targets are interfered with the least, calibrations are schedule at the closest time within a set interval. Maintenance and diagnostics are scheduled during intervals when no targets are available. When the station cannot range, it is set for maintenance (as indicated in the station mask file). Refer to figure below

Non-REAL TIME SCHEDULING



9.9) Focusing system using a star image

This algorithm adjusts the focus setting until a one arcsecond star image is obtained, indicating suitable focus. The focal metric (FM) measures the intensity of starlight within a certain radius of the centroid versus starlight outside that radius. It is defined quantitatively in the figure and equations in figure 9.9. If FM is positive then the focus criterion is met. If this criterion can not be met, then the focus setting is adjusted to give the best obtainable image.

Procedure:

The reference star catalog is searched [flowchart step 1] to find a bright star (M < 3) near the zenith (Z < 20 degrees). The telescope is slewed to the location of the star [2] and, while it is in motion, the exposure duration for the star is computed [3]. After the telescope settles at the star's location, the CCD camera is cleared, exposed, and read out [4]. The position of the centroid of the star is computed [5], and a small slew is executed (if needed) to center the star on the CCD frame [6]. The main loop of the function begins. If this is not the first time through the loop then current FM is saved as the old focal metric (OFM) [7]. The camera is cleared, exposed, and read out [8]. The new FM is computed [9]. If FM is positive then the focus criterion is met and the function returns a `1' [10]. Otherwise, the `direction of travel' table is consulted [11]. If FM is not greater than OFM and this is not the first time through the loop, or if the focus setting indicates that a focus limit has been hit, then the direction of focus is reversed. If a limit was hit then the appropriate `limit' flag is set, and if the direction has reversed then the `reverse' counter is incremented. If each focus limit was hit [12] this indicates that focusing has been completely unsuccessful, so the focus is set to the middle of the range and the function returns a `-1'. If the direction of travel has reversed twice [13] this implies that the best focus has been achieved but that the criterion has not been met, and the function returns a `0'. If the end of the loop is reached then the focus setting is adjusted [14], and the loop is executed again.

1) Select bright stars near zenith

2) Slew to star

3) Computer exposure duration

4) Expose and readout CCD

5) Centroid star image

6) Center the star

7) Save old focal metric (OFM)

8) Expose and read out CCD

9) Compute focal metric (FM)

L
O
O
P

10) If FM>0 RETURN 1

11) Consult Direciton of Travel Table

12) If 2 HFLs: Set, RETURN -1

13) If Reverse =2 RETURN 0

14) Adjust focuse setting

Pixels Around Centroid
(scale = 0.5 arcsec/pixel)

S	**S**						**S**	**S**
S			**3**	**3**	**3**			**S**
		3	**2**	**2**	**2**	**3**		
	3	**2**	**1**	**1**	**1**	**2**	**3**	
	3	**2**	**1**	**0**	**1**	**2**	**3**	
	3	**2**	**1**	**1**	**1**	**2**	**3**	
		3	**2**	**2**	**2**	**3**		
S			**3**	**3**	**3**			**S**
S	**S**						**S**	**S**

DNs are data numbers or pixel values

$DN0 = \Sigma DN(0)$	centered DN
$DN1 = \Sigma DN(1's)/8$	1 pixels from center
$DN2 = \Sigma DN(2's)/12$	2 pixels from center
$DN3 = \Sigma DN(3's)/16$	3 pixels from center
$DNS = \Sigma DN(S's)/12$	sky background

DS0 = DN0 - DNS
DS1 = DN1 - DNS
DS2 = DN2 - DNS
DS3 = DN3 - DNS
Focal Metric = 2 * DS0 + DS1 - DS2 - 2 * DS3

Direction of Travel Table

		Hit Focus Limit (HFL)	
		YES	NO
FM > OFM	YES	Reverse	Same
or 1st time	NO	Reverse	Reverse

Figure 9.9

9.10) Mount model

Since the mount axes are not a perfect coordinate system, the non-orthogonalities, misalignments and sag must be modeled. The simple 10-element model shown here models normal errors seen in an AZ-EL telescope mount. The mount errors (ΔAZ and ΔEL) are used to correct the mount encoder readings, and are functions of the mount azimuth (AZ) and elevation (EL). The coefficients of this model are obtained by pointing to enough stars to make a good statistical estimate (typically 2 to 3 times the number of coefficients) and obtaining the biases required to center each star in the receiver field of view. A least squares fit of the model to this data gives the coefficients.

$$\Delta AZ = A1+A2\cdot\sin(AZ)\cdot\tan(EL)+A3\cdot\cos(AZ)\cdot\tan(EL)+A4\cdot\tan(EL)+A5\cdot\sec(EL)$$
$$\Delta EL = B1+B2\cdot\cos(AZ)+B3\cdot\sin(AZ)+B4\cdot\cos(EL)+B5\cdot\sin(EL)$$

An eleventh term can and may be added to correct for refraction errors. This may be necessary for acquisition at low elevations. It is unclear at this point if this model will be sufficient for the SLR2000 telescope mount. At issue is both the number and contribution of the terms, and whether the coefficients can be fixed or must be a function of temperature (or some other independent variable).

9.11) Point Ahead / Behind

Since the speed of light is not infinity, the time from the laser fire to when the light reaches the satellite can be anywhere from a few milliseconds for low earth orbiting spacecraft to a sizeable fraction of a second for GPS and Etalon. The satellite will continue to move while the laser pulse is in flight and so the tracking station must lead the spacecraft (or point ahead of it) much like a quarterback leads his receiver in football. This is not a new concept; SLR stations have been doing point-ahead for a decade or more. What is new, however, is that the return problem, or point-behind has now also become an issue for SLR2000. This is due to the narrow receiver field of view that SLR2000 will be using. As with point-ahead the light reflecting off of the spacecraft corner-cubes will take time to return to the telescope. During this period the spacecraft will have moved and the light will, of course, appear to come from the spacecraft's position at the time of reflection. Angularly the difference between the point-ahead and point-behind is shown in the following table.

Satellite	Elevation = 20deg	Elevation =90deg
STARLETTE	6 arcsec	11 arcsec
LAGEOS	7 arcsec	8 arcsec
ETALON	5 arcsec	5 arcsec

The field of view of the SLR2000 receiver is expected to have a radius of around 5 arcseconds. Since the telescope must be pointed ahead for each fire, it cannot at the same time point-behind to also receive the return, and as seen in the table above, for most spacecraft this means that the return will not fall within the receiver field of view.

There are two ways to correct for this: (1) make the receiver field of view larger, or (2) add an extra turning mirror that can be controlled by the computer. Each has advantages and disadvantages. The larger field of view is simpler in that it has no moving parts but the larger FOV adds a lot of background noise and also adds complexity for the software since the software will have to drive the telescope to put the returns at a spot no longer centered on the quadrant detector. This requires an extremely good knowledge of the quadrant detector scale factors, the image size on the detector, and the image rotation algorithm, and relies solely on the theory for the desired location of the return in order to optimize the transmit pointing. On the other hand, the turning mirror adds a moving part that must be controlled by the computer, and the angle of this mirror is determined solely by the theory. However, it allows the computer to drive the return to the center of the quadrant detector and eliminates all of the problems associated with off-center guiding.

The point-ahead / point-behind angles are computed from the following.

$$\begin{gathered}R = R(t); \quad \Delta t = R/c \\ AZ_{ahead} = Z_{\alpha}(X(t+\Delta t),t) \\ EL_{ahead} = Z_{\epsilon}(X(t+\Delta t),t) \\ AZ_{behind} = Z_{\alpha}(X(t-\Delta t),t) \\ EL_{behind} = Z_{\epsilon}(X(t-\Delta t),t)\end{gathered}$$

where t is the current time, R is the range at time “t” , X is the geocentric rectangular satellite position vector, (AZ_{ahead},EL_{ahead}) are the laser transmit angles required to center the outgoing laser pulse on the satellite, and (AZ_{behind},EL_{behind}) are the apparent angles of the incoming return light. Z represents the transformation required to turn the inertial geocentric rectangular satellite position into local apparent angles and range. In this transformation the second variable in Z(·,t) implies that the calculation of local sideral time is computed at time “t”.

The angular difference between point-ahead and point-behind is thus:

$$\Delta\theta = \sqrt{(AZ_{ahead} - AZ_{behind})^2 \cos(EL)^2 + (EL_{ahead} - EL_{behind})^2}$$

9.12) Image Rotation:

Due to the path the light will be taking through the Coude optics, images will rotate as the mount azimuth and elevation change. This rotation will have to be removed via the following transformation for the star camera and the quadrant detector:

$$\begin{gathered}x^{/} = x\cos(\theta) + y\sin(\theta) \\ y^{/} = -x\sin(\theta) + y\cos(\theta) \\ where \;\; \theta = AZ + EL + constant\end{gathered}$$

where the value of θ will probably change when the optical head design is finalized. Here (x,y) is the image distance from the center of the field of view in the camera or quadrant detector coordinate system.

10)POP Flowcharts

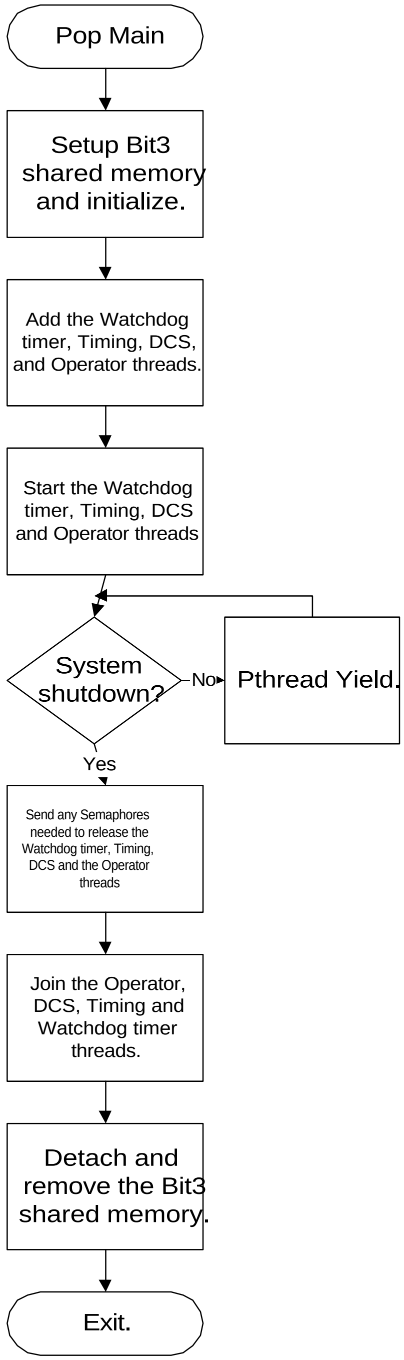


Figure 10.1

jwc:08/19/98

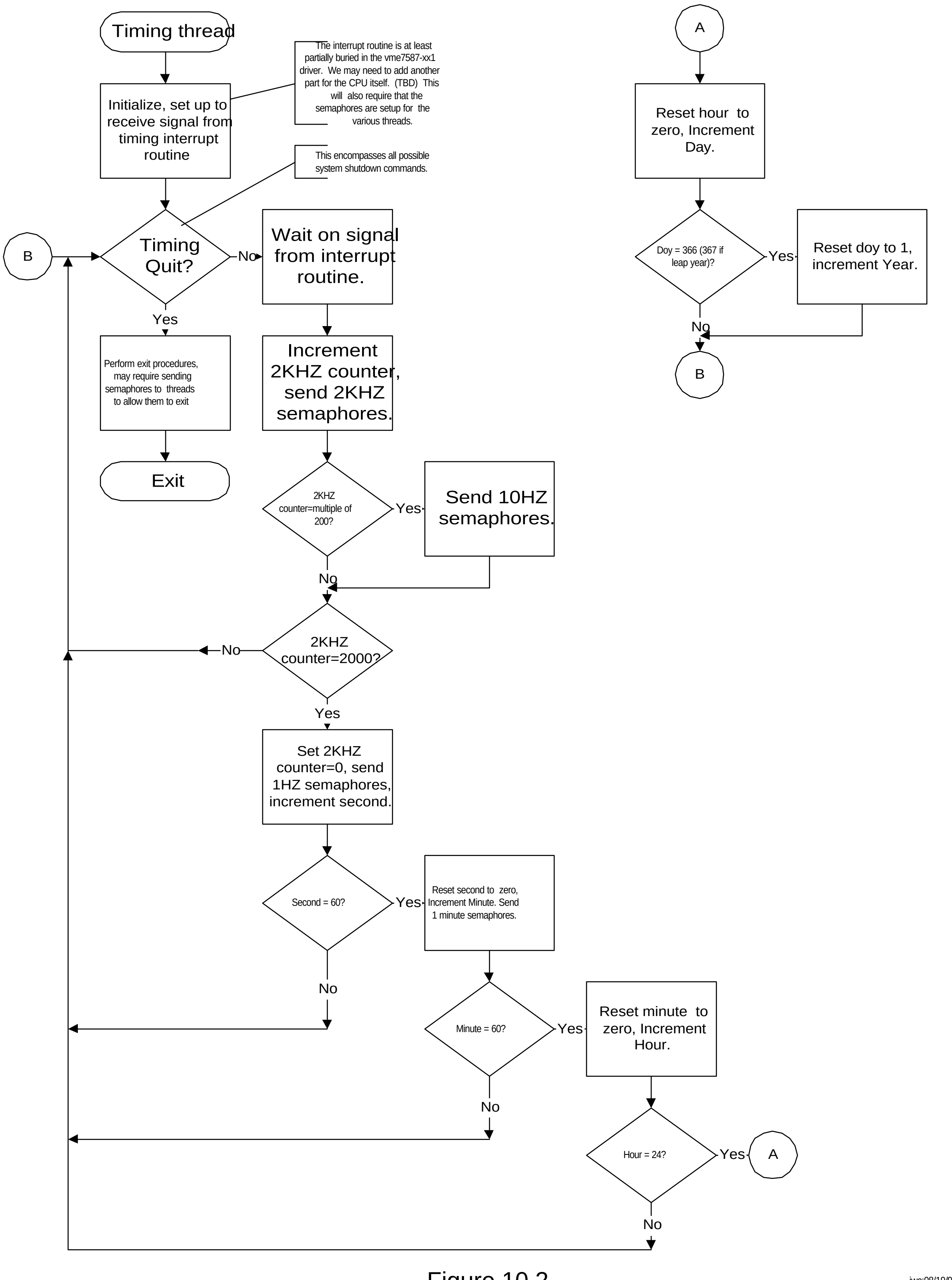
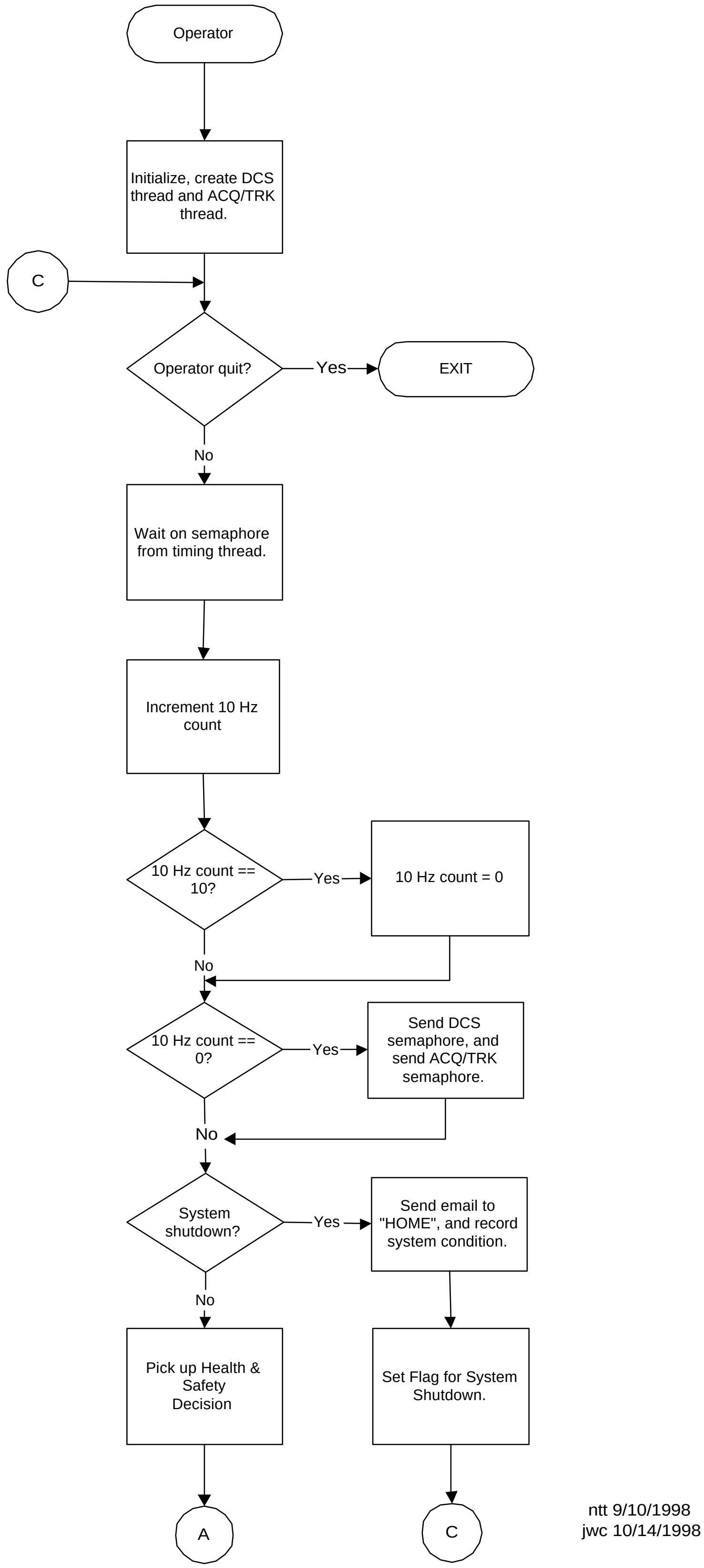

Timing thread
Initialize, set up to receive signal from timing interrupt routine
The interrupt routine is at least partially buried in the vme7587-xx1 driver. We may need to add another part for the CPU itself. (TBD) This will also require that the semaphores are setup for the various threads.
This encompasses all possible system shutdown commands.
B
Timing Quit?
No
Yes
Wait on signal from interrupt routine.
Perform exit procedures, may require sending semaphores to threads to allow them to exit
Exit
Increment 2KHZ counter, send 2KHZ semaphores.
2KHZ counter=multiple of 200?
Yes
Send 10HZ semaphores.
No
2KHZ counter=2000?
No
Yes
Set 2KHZ counter=0, send 1HZ semaphores, increment second.
Second = 60?
Yes
Reset second to zero, Increment Minute. Send 1 minute semaphores.
No
Minute = 60?
Yes
Reset minute to zero, Increment Hour.
No
Hour = 24?
Yes
A
No
A
Reset hour to zero, Increment Day.
Doy = 366 (367 if leap year)?
Yes
Reset doy to 1, increment Year.
No
B
jwc:08/19/98

Figure 10.2



Operator
Initialize, create DCS thread and ACQ/TRK thread.
C
Operator quit?
Yes
EXIT
No
Wait on semaphore from timing thread.
Increment 10 Hz count
10 Hz count == 10?
Yes
10 Hz count = 0
No
10 Hz count == 0?
Yes
Send DCS semaphore, and send ACQ/TRK semaphore.
No
System shutdown?
Yes
Send email to "HOME", and record system condition.
No
Pick up Health & Safety Decision
Set Flag for System Shutdown.
A
C
ntt 9/10/1998
jwc 10/14/1998

Figure 10.3a

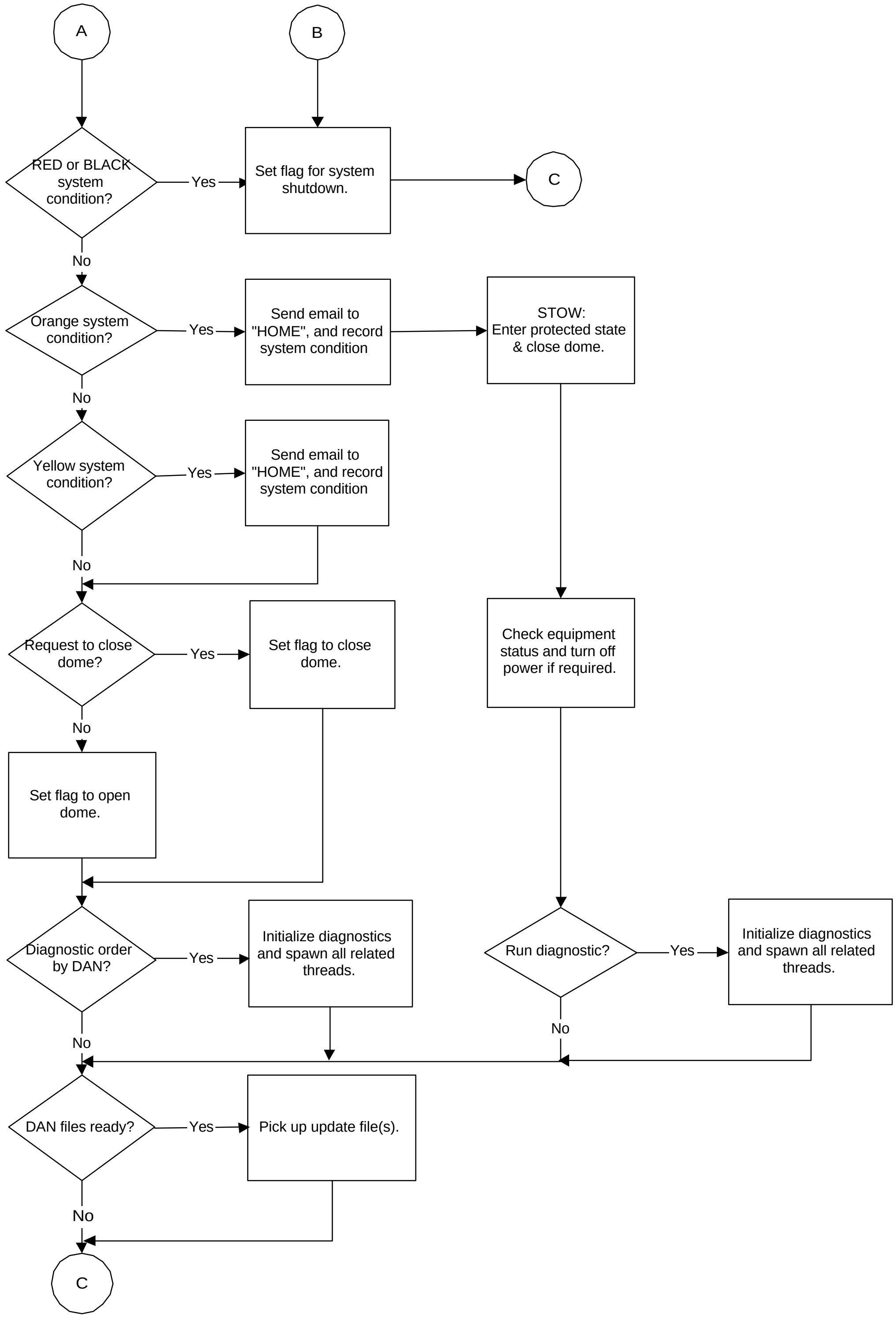


figure 10.3b

ntt 8/12/98

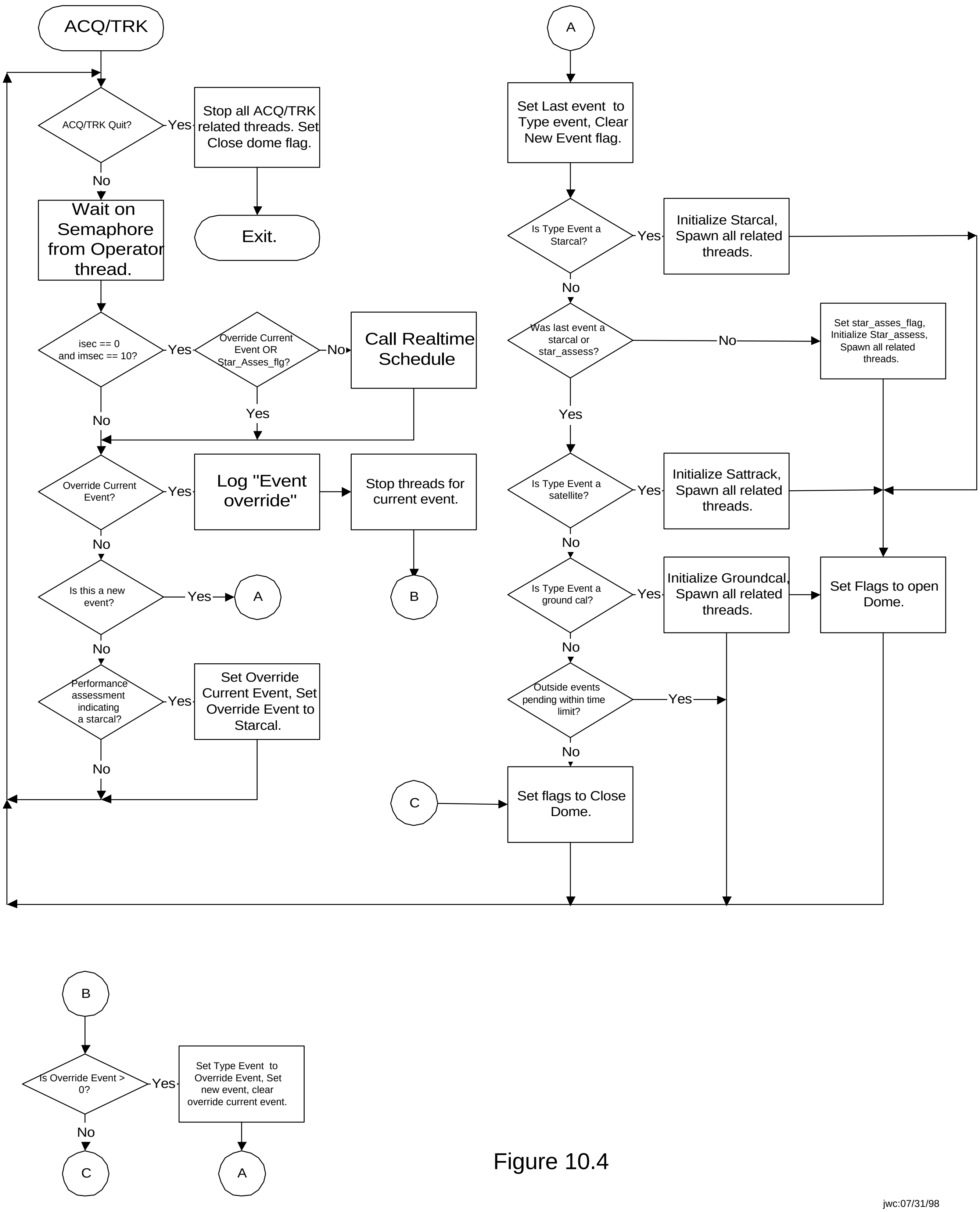

ACQ/TRK
ACQ/TRK Quit?
Yes
Stop all ACQ/TRK related threads. Set Close dome flag.
Exit.
No
Wait on Semaphore from Operator thread.
isec == 0 and imsec == 10?
Yes
Override Current Event OR Star_Asses_flg?
No
Call Realtime Schedule
Yes
No
Override Current Event?
Yes
Log "Event override"
Stop threads for current event.
No
Is this a new event?
Yes
A
B
No
Performance assessment indicating a starcal?
Yes
Set Override Current Event, Set Override Event to Starcal.
No
A
Set Last event to Type event, Clear New Event flag.
Is Type Event a Starcal?
Yes
Initialize Starcal, Spawn all related threads.
No
Was last event a starcal or star_assess?
No
Set star_asses_flag, Initialize Star_assess, Spawn all related threads.
Yes
Is Type Event a satellite?
Yes
Initialize Sattrack, Spawn all related threads.
No
Is Type Event a ground cal?
Yes
Initialize Groundcal, Spawn all related threads.
Set Flags to open Dome.
No
Outside events pending within time limit?
Yes
No
C
Set flags to Close Dome.
B
Is Override Event > 0?
Yes
Set Type Event to Override Event, Set new event, clear override current event.
No
C
A
jwc:07/31/98

Figure 10.4

REAL TIME SCHEDULING THREAD

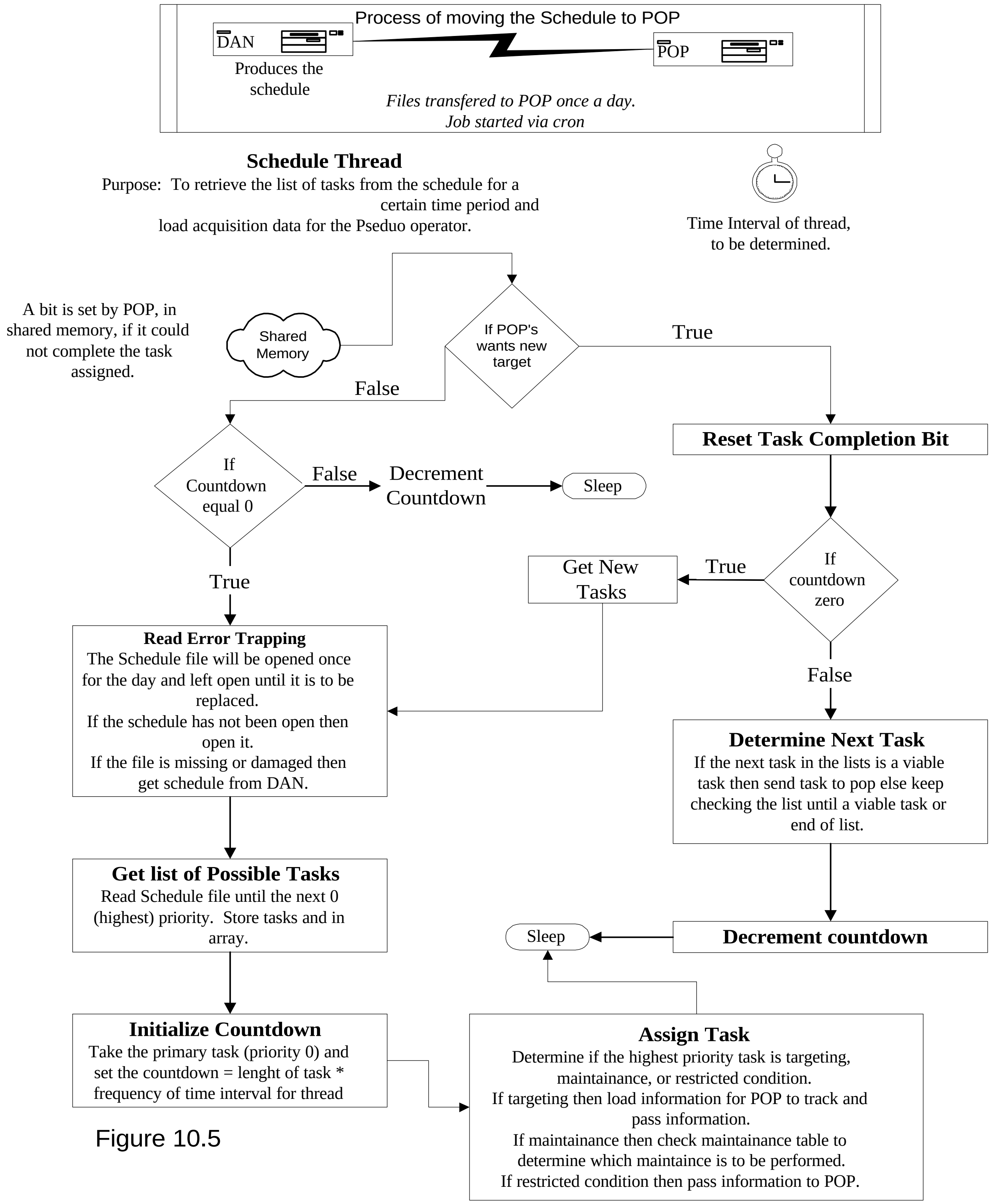


Figure 10.5

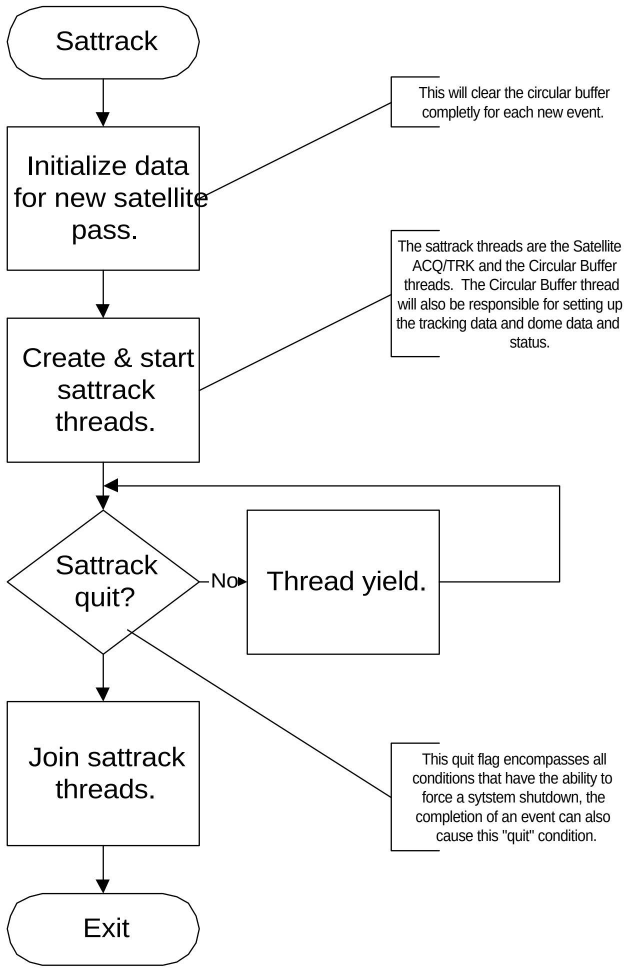


Figure 10.6

jwc:08/13/98

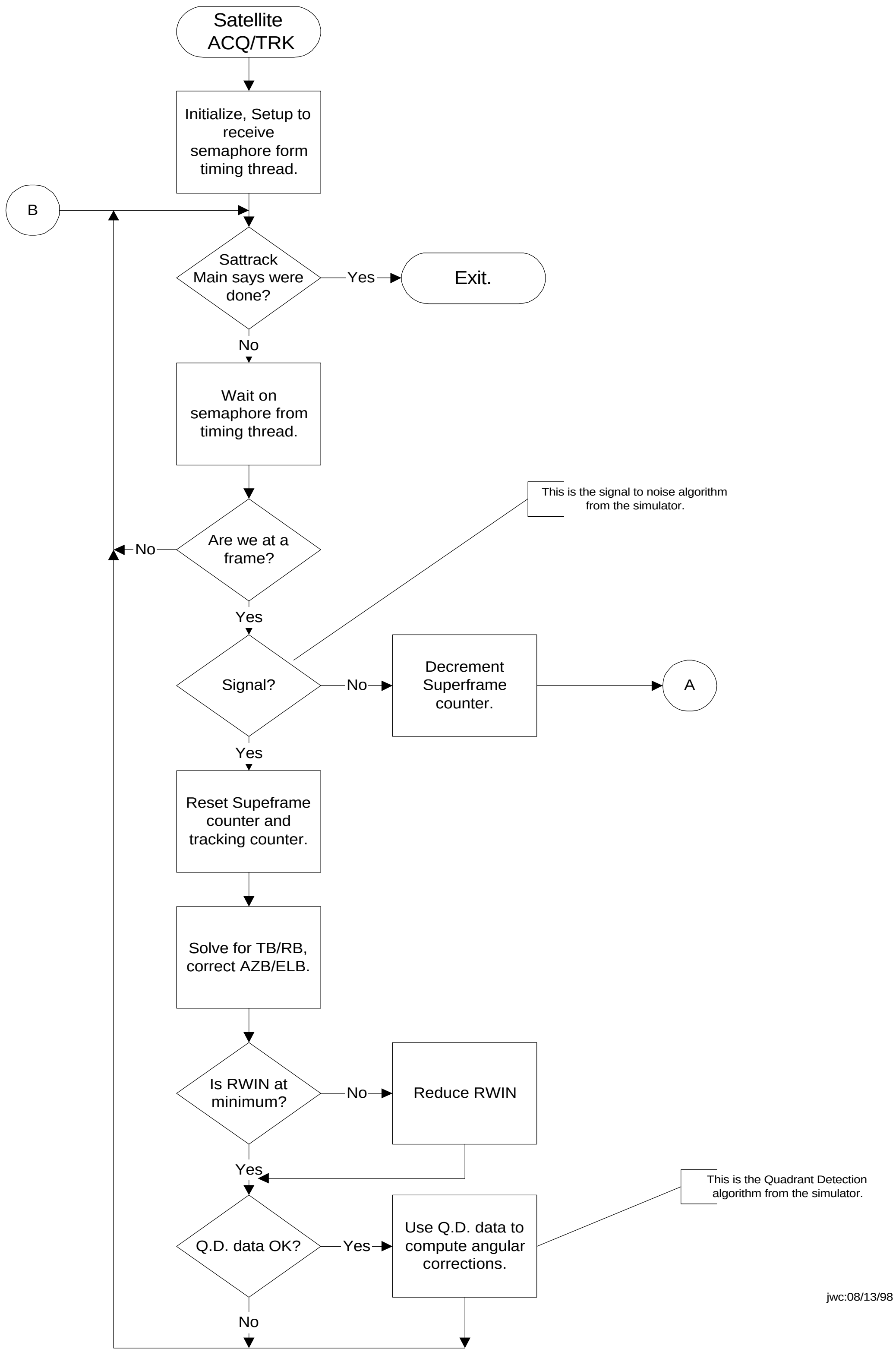

Satellite ACQ/TRK
Initialize, Setup to receive semaphore form timing thread.
B
Sattrack Main says were done?
Yes
Exit.
No
Wait on semaphore from timing thread.
This is the signal to noise algorithm from the simulator.
Are we at a frame?
No
Yes
Signal?
No
Decrement Superframe counter.
A
Yes
Reset Supeframe counter and tracking counter.
Solve for TB/RB, correct AZB/ELB.
Is RWIN at minimum?
No
Reduce RWIN
Yes
This is the Quadrant Detection algorithm from the simulator.
Q.D. data OK?
Yes
Use Q.D. data to compute angular corrections.
jwc:08/13/98
No

Figure 10.7a

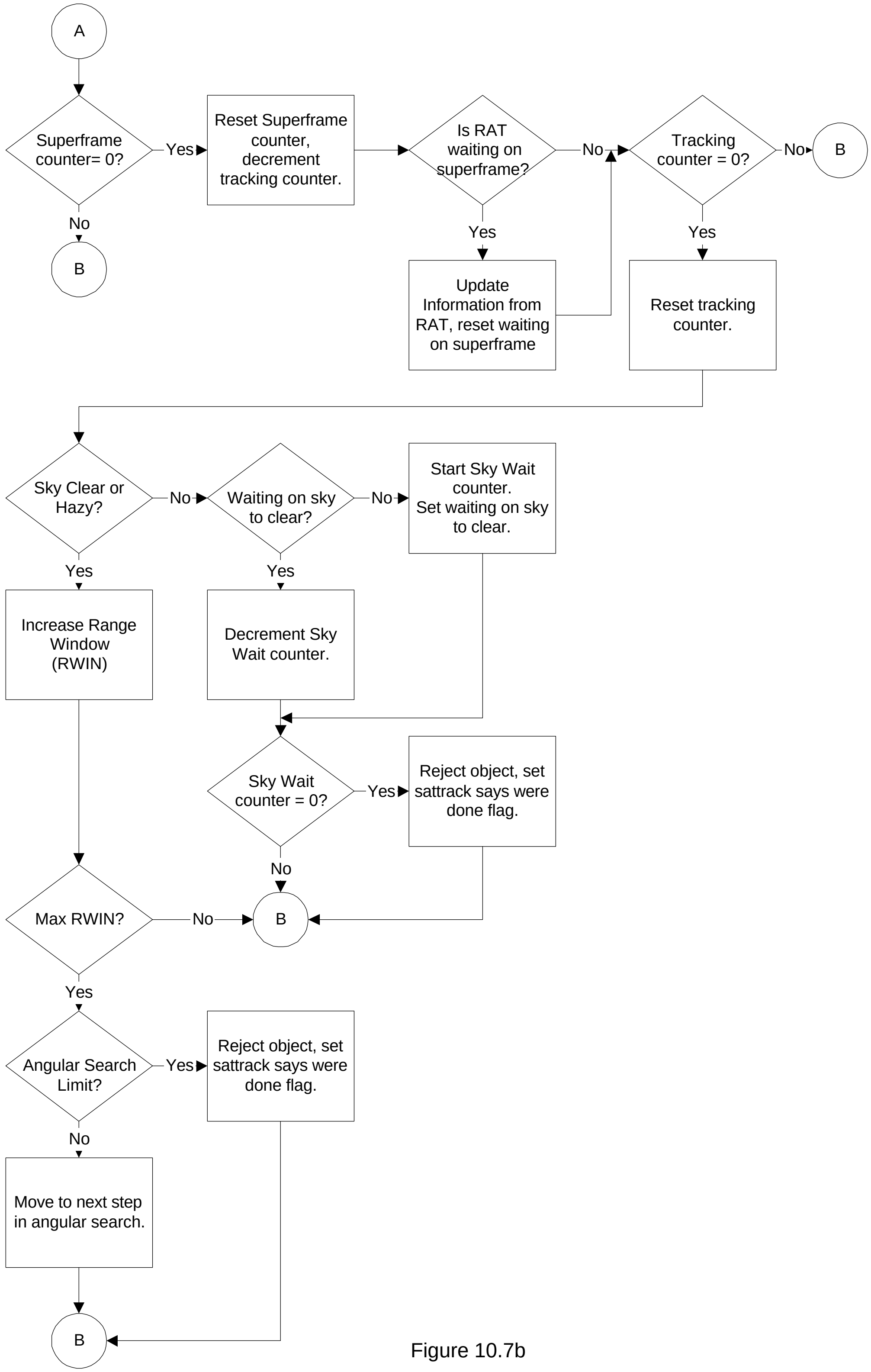


Figure 10.7b

jwc:08/13/98

ACQUISITION RANGE & ANGULAR SEARCH

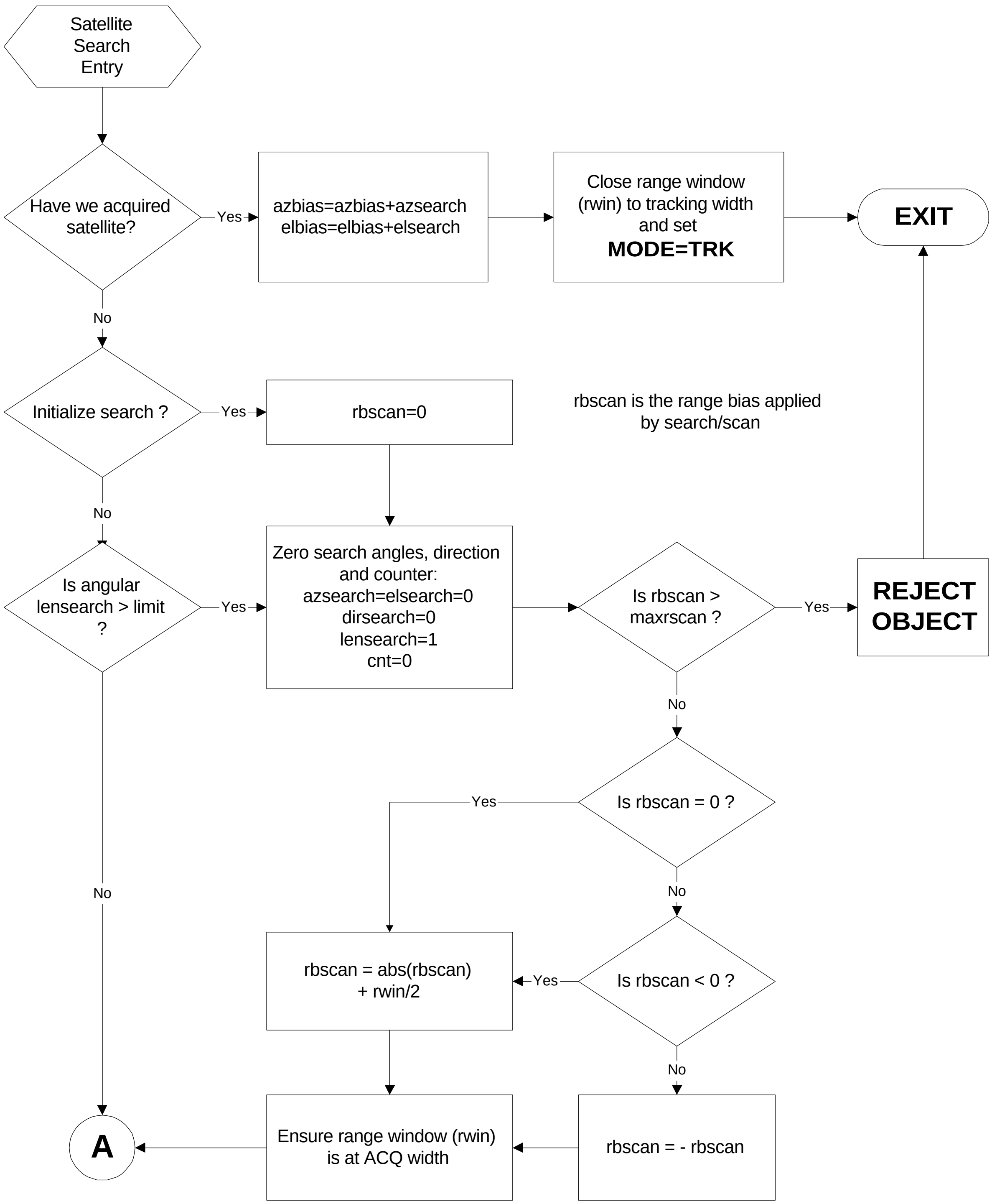


Figure 10.8a

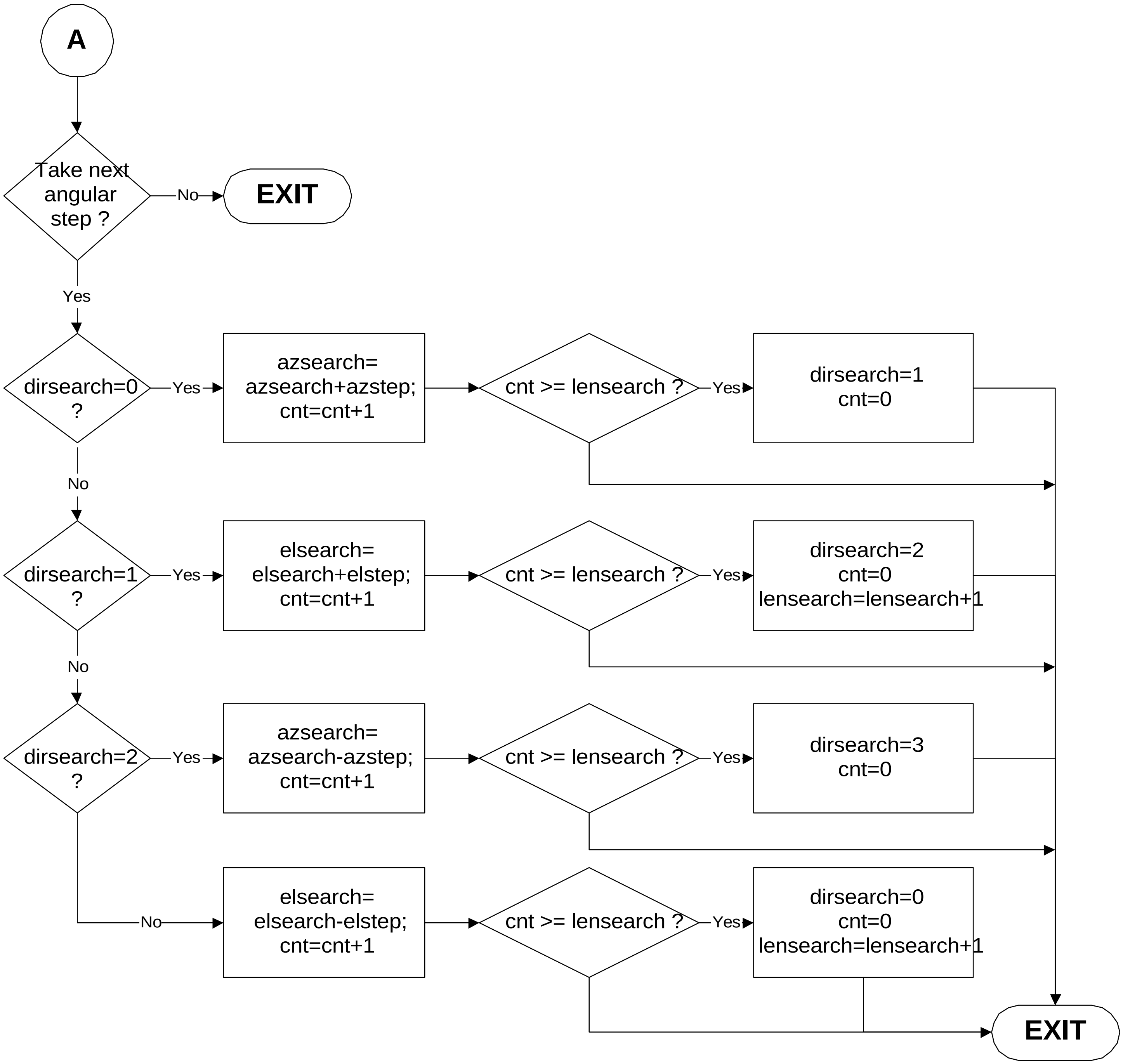

A
Take next angular step ?
No
EXIT
Yes
dirsearch=0 ?
Yes
azsearch= azsearch+azstep; cnt=cnt+1
cnt >= lensearch ?
Yes
dirsearch=1 cnt=0
No
dirsearch=1 ?
Yes
elsearch= elsearch+elstep; cnt=cnt+1
cnt >= lensearch ?
Yes
dirsearch=2 cnt=0 lensearch=lensearch+1
No
dirsearch=2 ?
Yes
azsearch= azsearch-azstep; cnt=cnt+1
cnt >= lensearch ?
Yes
dirsearch=3 cnt=0
No
elsearch= elsearch-elstep; cnt=cnt+1
cnt >= lensearch ?
Yes
dirsearch=0 cnt=0 lensearch=lensearch+1
EXIT

Figure 10.8b

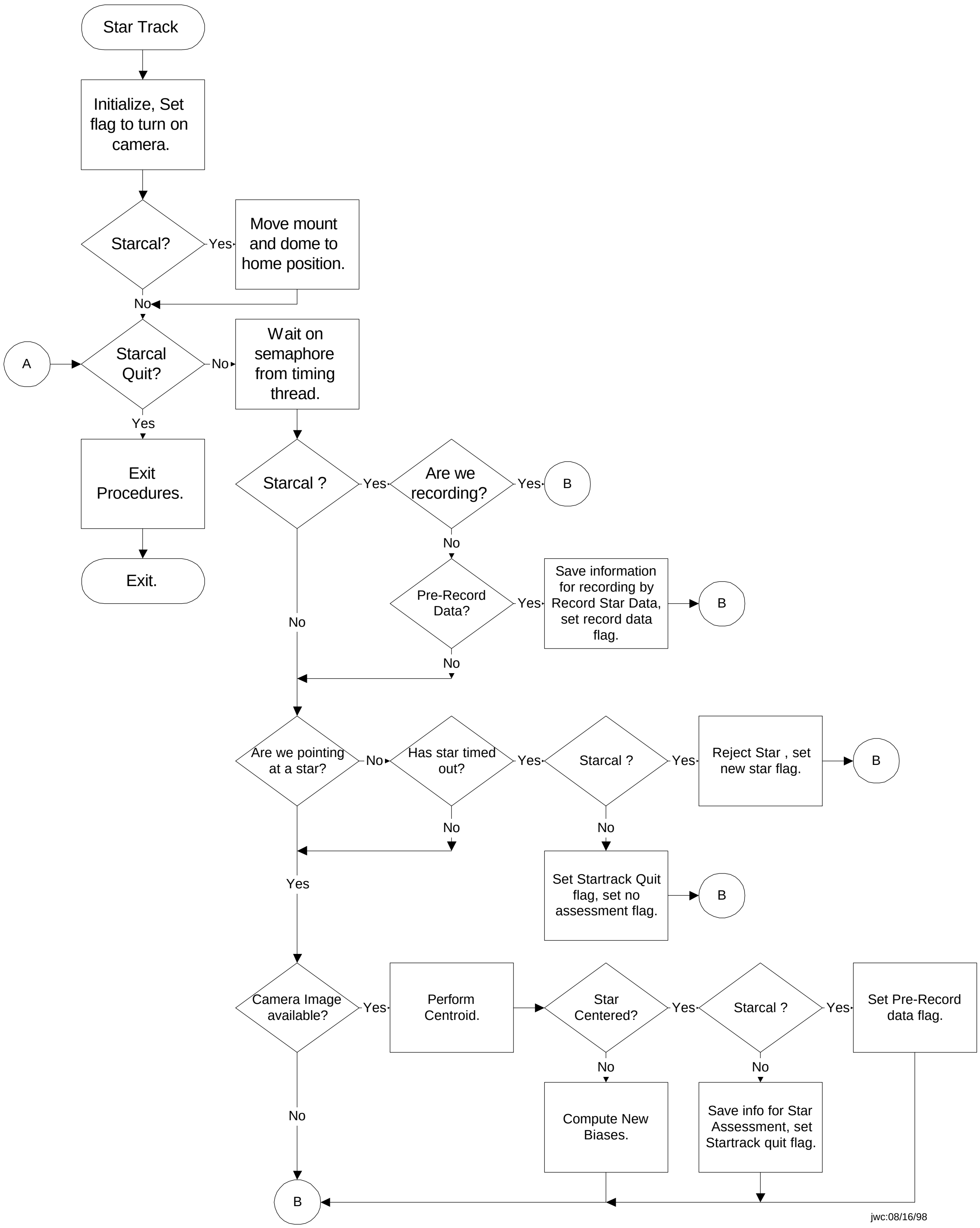

Star Track
Initialize, Set flag to turn on camera.
Starcal?
Yes
Move mount and dome to home position.
No
A
Starcal Quit?
No
Wait on semaphore from timing thread.
Yes
Exit Procedures.
Exit.
Starcal ?
Yes
Are we recording?
Yes
B
No
Pre-Record Data?
Yes
Save information for recording by Record Star Data, set record data flag.
B
No
No
Are we pointing at a star?
No
Has star timed out?
Yes
Starcal ?
Yes
Reject Star , set new star flag.
B
No
No
Yes
Set Startrack Quit flag, set no assessment flag.
B
Camera Image available?
Yes
Perform Centroid.
Star Centered?
Yes
Starcal ?
Yes
Set Pre-Record data flag.
No
No
Compute New Biases.
Save info for Star Assessment, set Startrack quit flag.
No
B
jwc:08/16/98

Figure 10.9a

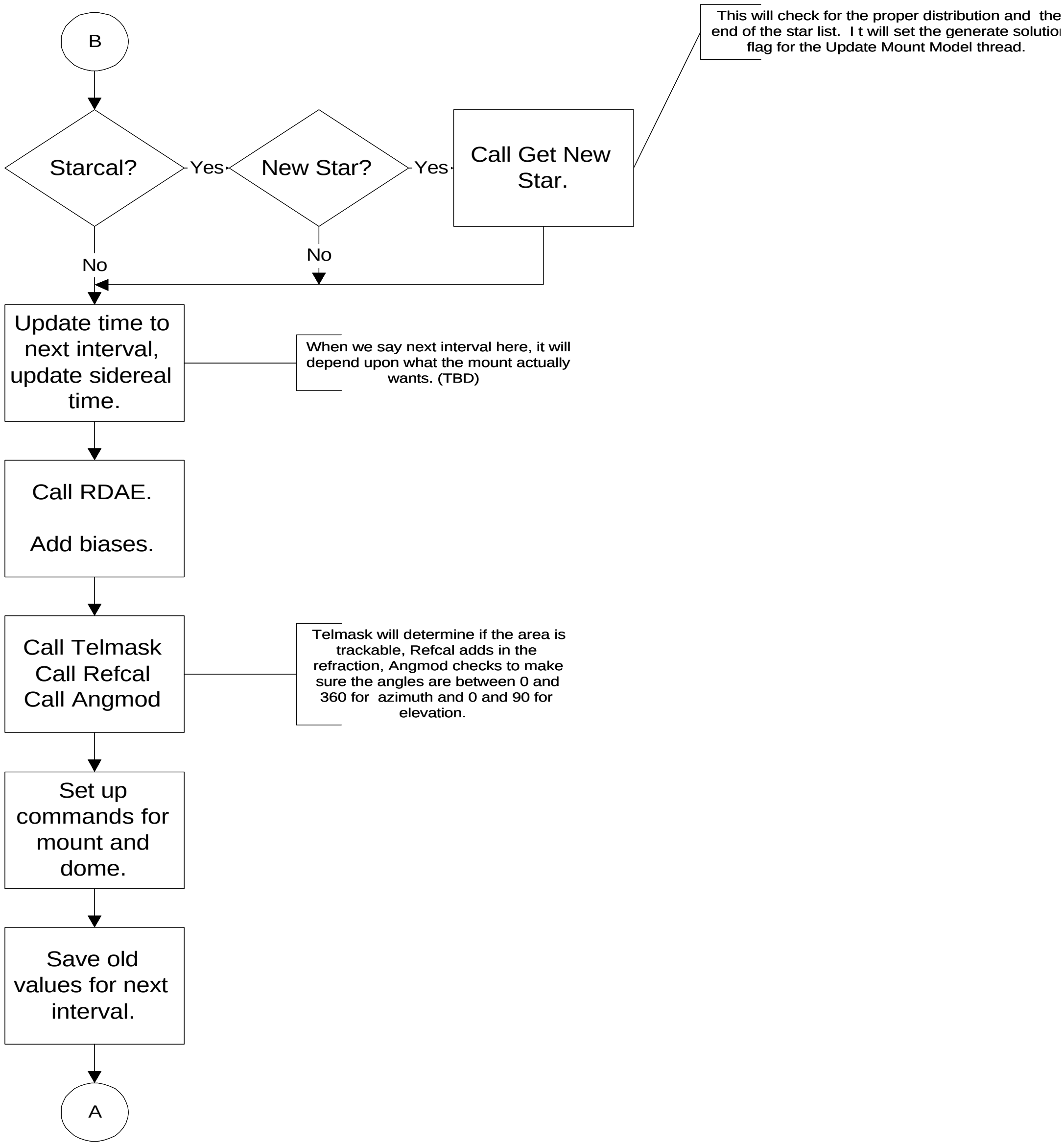


Figure 10.9b

jwc:08/16/98

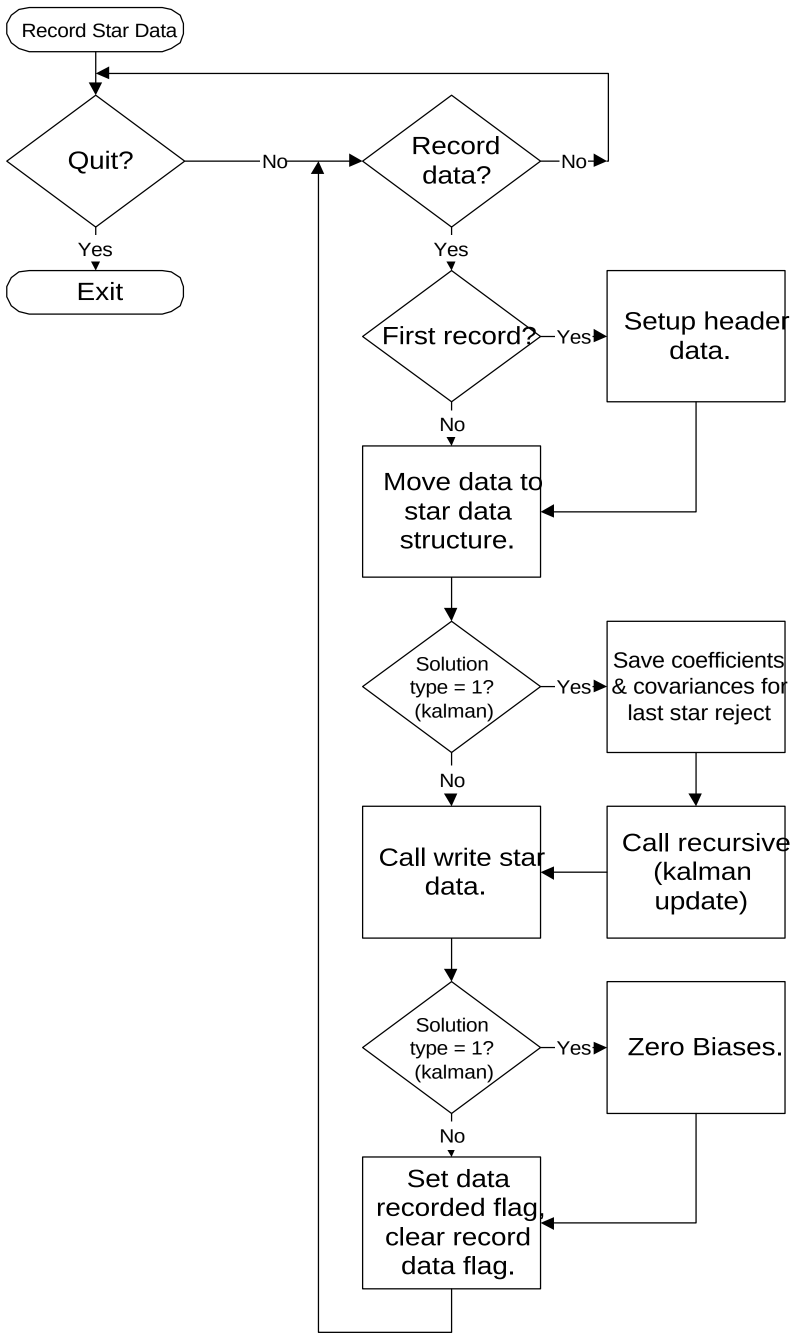


Figure 10.10

jwc:08/16/98

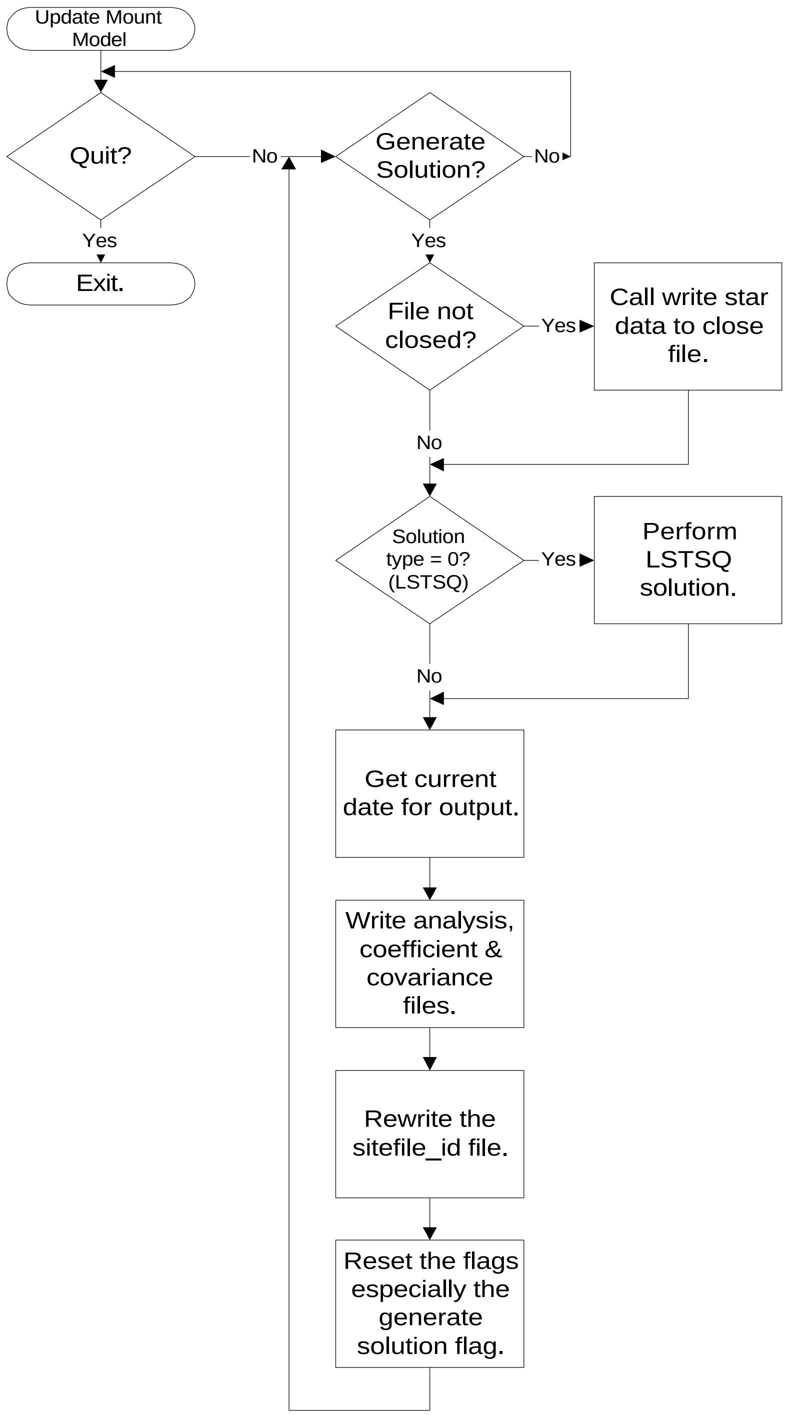


Figure 10.11

jwc:08/16/98

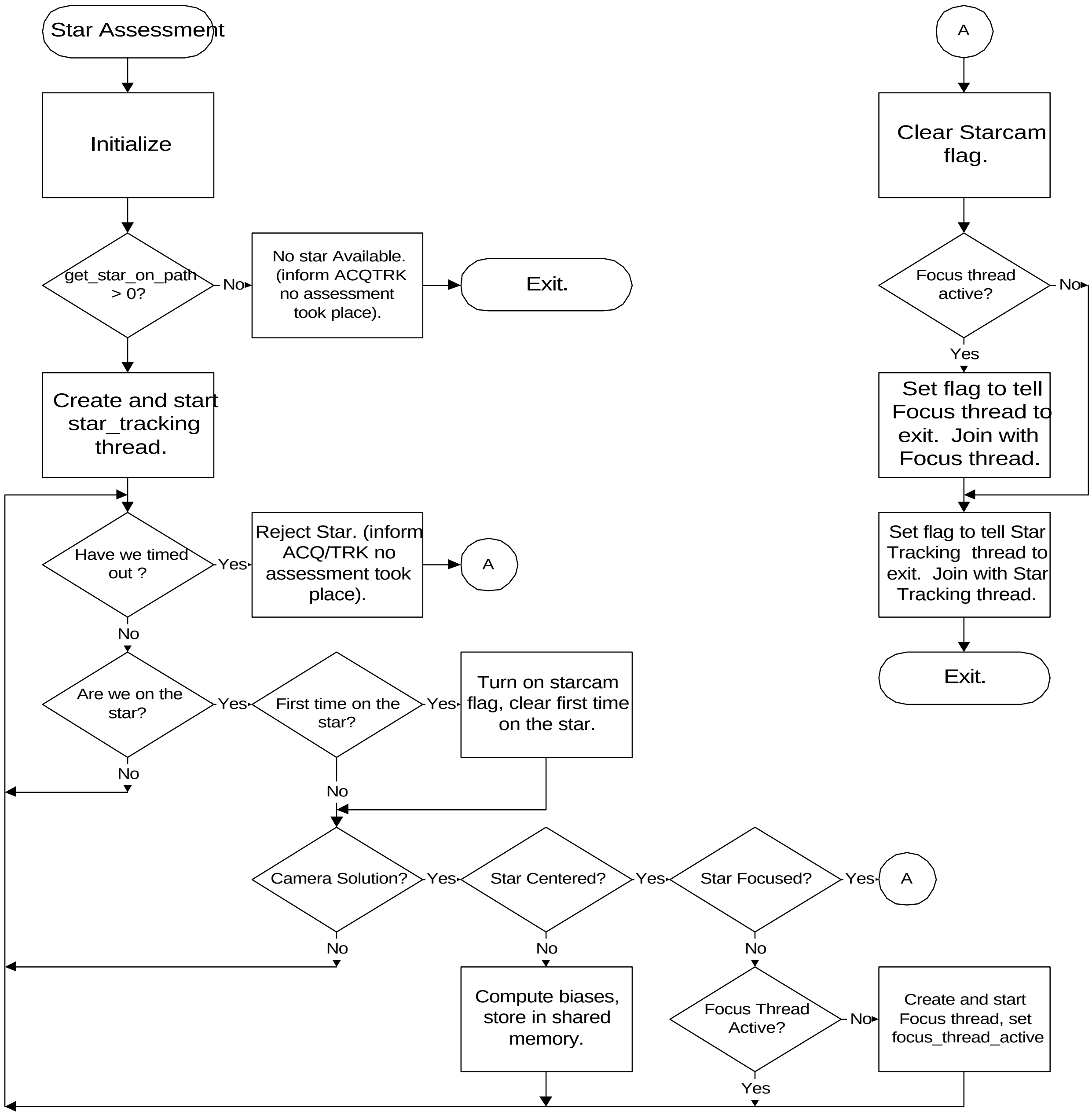


Figure 10.12

jwc:08/16/98

11) Data flow / Communication

11.1) Data flow / Communication ICC↔ POP

11.1.1) ICC → POP

Data transfer via shared memory (see iccpopshm.h, section 13)

11.1.2) POP → ICC

Data transfer via shared memory (see iccpopshm.h, section 13)

11.2) Data flow / Communication POP↔DAN

11.2.1) POP → DAN

11.2.1.1) Operational log FILE

Normal is log only signal returns
For testing can log all returns (noise too)
Includes weather information, sky and dome

11.2.1.2) Data transfer via shared memory (see popdanshm.h , section 13)
11.2.1.3) Star camera file
11.2.1.4) Star summary file

11.2.2) DAN → POP

11.2.2.1) Predictions

File of prediction information (vectors every minute) that resides on disk, is updated by DAN daily. POP determines if file is updated by a flag in shared memory.

11.2.2.2) Schedule

File of schedule information that resides on disk, is updated by DAN daily. POP determines if file is updated by a flag in shared memory.

11.2.2.3) Satellite related data

Information required for each satellite in order to :
1) process predicts
2) search and find satellite
3) determine signal from noise

11.2.2.4) Station File

This file includes the station specification such as latitude, longitude, height, name and ID, azimuth and elevation offsets. It also Includes starting operational date for the system in current location and other system parameters.

11.2.2.5) Real time flags & data :
Data transfer via shared memory (see popdanshm.h, section 13)

11.2.2.6) Simulation and test parameters (Via shared memory from rant and/or via file)

11.2.2.7) Emergency Shutdown Flag: Shared Memory
Flag to indicate immediate system shutdown required.
Flag to indicate immediate turn off of laser.
Flag to indicate immediate closure of dome (STOW).

These flags passed via shared memory (see popdanshm.h, section 13)

11.2.2.8) Mask file

11.2.2.9) Maintenance file

11.2.2.10) Target Database file

11.3) Data flow / Communication DAN ↔ E.F.

11.3.1) DAN → E.F.

CSTG Normal Point file
CSTG Engineering file
Daily Diary
Weather Data File
Emergency shut down message
Emergency contacts

11.3.2) E.F. → DAN
Schedule input file (Satellite priority file!!)
Time bias
Satellite Predictions
IERS.DUT

11.4) Data flow / Communication DAN↔ RAT

11.4.1) DAN →RAT

Schedule file
Security camera file
Sky camera file
Health and safety file
Star camera file
Star summary file
Data transfer via shared memory (see danpopshm.h , section 13)
Data transfer via shared memory (see ratshm.h, section 13)

11.4.2) RAT➙ DAN

Schedule file
Data transfer via shared memory (see ratshm.h, section 13)
Simulation and test parameters (via ratshm.h)

11.5) Data flow / Communication POP ↔ RAT

11.5.1)POP ➙ RAT

11.5.1.1) Circular buffer debug file (only turned on during testing): FILE

Range Circular Buffer
Angular Circular Buffer

11.5.1.2) Message file

11.5.1.3) site_file_ID

11.5.1.4) Mount Model file

11.5.2) RAT ➙ POP

11.5.2.1) Simulation and test parameters

Data transfer via shared memory (see ratshm.h, section 13)

12) Formats

Table of contents

Calibration Database Capture File Format (CCR)

FIELD	BYTES	DESCRIPTION	FORMAT
1	01-19	Station-Date-Time-Target	A19
2	20-22	Occupation Number	A3
3	23-31	Target Distance (mm)	F9.0
4	32-37	Time of First Observation	I6
5	38-43	Cal Set Observations	I6
6	44-49	Cal Set Rejects	I6
7	50-54	Cal Set RMS (mm)	F5.1
8	55-62	Cal System Delay (mm)	F8.1
9	63-67	Cal Linear Term	F5.2
10	68-72	Cal Al Mean	F5.2
11	73-77	Cal Al RMS	F5.2
12	78-82	Cal PMT Mean (v)	I5
13	83-87	Cal PMT RMS (v)	F5.2
14	88-91	Cal Transmit Mean	I4
15	92-98	Cal Transmit RMS	F7.2
16	99-102	Cal Receive Mean	I4
17	103-109	Cal Receive RMS	F7.2
18	110-118	Cal Transmit Delay Mean (ms)	F9.2
19	119-127	Cal Transmit Delay RMS (ms)	F9.2
20	128-134	Minimum System Delay (m)	F7.0
21	135-141	Maximum System Delay (m)	F7.0
22	142-144	Calibration Sigma Multiplier (.1)	I3
23	143-144	Processor Version	A2
24	145-146	Addition Number	I2
25	147-152	Processor Run Rate	I6
26	153-158	Mean Temperature (0.01° C)	I6

CONTROLLER RANGE DATA FORMATS
System Control

	UNITS VALUE	CHARS. SCALE	FMT.
Line 0			
Section/Line Identifier	10	XX	A2
Data Identifier	ASCII	CONTROLLER	A12
Line 1			
Section/Line Identifier	11	XX	A2
Control Name (See Note 1)	ASCII	‘ID’	A12

	ID
Note 1: Satellite Data	SATELLITE
Lunar Data	LUNAR
Aircraft Data	AIRCRAFT
Simulation Data	SIMULATE
Clear Tracking Scenario (Calb only)	CTS/CAL
Clear Tracking Scenario (Calb and Satl)	CTS/CAL/SAT
End of File	EOF
End of Scenario	EOS
Change Tracking Scenario	SCENARIO #i (i = 0.1 ... n)

Note 2 There are no decimal points in the floating point values.
The floating point values are scaled up by the number of decimals ** to the power of ten. this was necessary to circumvent a format problem (fprintf) in Lynx The format shows the implied decimal scaling. All values are right justified.

CONTROLLER RANGE DATA FORMATS

Site Information

	UNITS VALUE	CHAR. SCALE	FMT.
Line 0			
Section/Line Identifier	20	XX	A2
Data Identifier	ASCII	SITE INFO	A12
Line 1			
Section/Line Identifier	21	XX	A2
Site Identifier	00 - 99	XX	A3
Move #	00 - 99	XX	A3
Format #	00 - 99	XX	A3
System Configuration #	0 - 9	X	A2
Marker #	0 - 9's	XXXX	A5
Side occupation Desgn. #	0 - 9's	XXXXXXXX	A9
Monitor Version #	00 - 99	XX	A3
Program Version #	00 - 99	XX	A3
Wave Length#1	Angstroms	XXXXX	A6
Wave Length #2 (48-inch)	Angstroms	XXXXX	A6
USNO/Station Clk Offset	ns	SXXXXXXXX X	A11
Cesium/Station Clk Offset	ns	SXXXXXXXX SXXXXXXXX X	A11
Timera_Cycle_TU	ns		A10/77
Line 2			
Section/Line Identifier	22	XX	A2
Latitude, Geodetic	deg.	SXX.XXXXXX	A11
Longitude	deg.	XXX.XXXXXX	A11
Height, Geodetic	m	SXXXX.X	A8
X-Eccentricity (MLRS)	mm	SXXXX	A6
Y-Eccentricity (MLRS)	mm	SXXXX	A6
Z-Eccentricity (MLRS)	mm	SXXXX	A6
Azimuth Offset	deg.	SXXX.XXXX	A10
Elevation Offset	deg.	SX.XXXX	A8
Barometric Pres. Offset	mbar	SX.XX	A6
Temperature offset	deg.	SXX.X	A6/80
Line 3			
Section/Line Identifier	23	XX	A2
Transmit Optics Correction	mm	XX.X	A5
Translator Range Corr. - Xmit	mm	XXX.X	A6
Daylight Filter Range Corr.	mm	XX.X	A5
T/R Switch Range Corr. (T-3/4)	mm	XX.X	A5
Spectral filter (MLRS)	nm	X.XX	A5
Spatial Filter (MLRS)	arcsec	XX	A3/11

Line 4

Section/Line Identifier	24	XX	A2
Camera #3 Port Number	-	X	A2
TV Camera #1 Offset (48-inch)	pixels	XXX	2A4
TV Camera #2 Offset (48- inch)	pixels	XXX	2A4
TV Camera #3 Offset (48-inch)	pixels	XXX	2A4
TV Camera #1 Scale (48-inch)	deg/pixel	X.XXXX	2A7
TV Camera #2 Scale (48-inch)	deg/pixel	X.XXXX	2A7
TV Camera #3 Scale (48-inch)	deg/pixel	X.XXXX	2A7/70

Line 5

Section/Line Identifier	25	XX	A2
Number Model Coefficients	-	XX	A3
Vernier Slopes (MLRS)	ps/count	XXX.XXXX	8A9/77

Line 6 ...

Section/Line Identifier	26	XX	A2
Mount Model Coefficients	mdeg	SZZZ.XXXX	A10/12

Line 7

Section/Line Identifier	27	XX	A2
Dome Azimuth Offset	mdeg	XXX.XXXX	A8

CONTROLLER RANGE DATA FORMATS
Header Information

CALIBRATION	UNITS VALUE	CHARS. SCALE	FMT.
Line 0			
Section/Line Identifier	30	XX	A2
Data Identifier	A - Z	HEADER CALB	A12
Line 1			
Section/Line Identifier	31	XX	A2
Reference Time	Year	XXXX	A5
	DOY	XXX	A4
	Hour	XX	A3
	Minute	XX	A3
Target ID	(A - Z)	X	A2
Target Azimuth	Deg.	XXX.XXXX	A9
Target Elevation	Deg.	SXX.XXXX	A9
Target Range	m (1-way)	XXXX.XXXX	A10
Target Cross section	m*m	XXXXXX.X	A8/55

Note: Not applicable to MLRS

TEST

	UNITS VALUE	CHARS. SCALE	FMT.
Line 0			
Section/Line Identifier	40	XX	A2
Data Identifier	A - Z	HEADER TEST	A12
Line 1			
Section/Line Identifier	41	XX	A2
Reference Time	Year	XXXX	A5
	DOY	XXX	A4
	Hour	XX	A3
	Minute	XX	A3
Target ID	A - Z	X	A2
Target Azimuth	Deg.	XXX.XXXX	A9
Target Elevation	Deg.	SXX.XXXX	A9
Target Range	m (1-way)	XXXX.XXXX	A10
Target Cross section	m*m	XXXXXX.X	A8/55

Note: Not applicable to MLRS

SATELLITE

	UNITS VALUE	CHARS. SCALE	FMT.
Line 0			
Section/Line Identihier	50	xx	A2
Data Identifier	A-Z	HEADER SATL	A12

Line 1

Section/Line Identifier	51	XX	A2
Reference Time	Year	XXXX	A5
	DOY	XXX	A4
	Hour	XX	A3
	Minute	XX	A3
SIC #	-	XXXX	A5
Ephemeris #	-	XX	A3
Time of Vector (PCA)	Hour	XX	A3
	Minute	XX	A3
	Second	XX	A3
	DOY	XXX	A4
	Year	XXXX	A5
X-Polar Motion	arc-sec	XXXX	A5
Y-Polar Motion	arc-sec	XXXX	A5
Target Crossection	m*m	XXXXXXXXX X	A11/64

Line 2

Section/Line Identifier	52	XX	A2
Position Vector (PCA)	m	SXXXXXXX.X	3A12
Velocity Vector (PCA)	m/sec	X SXXXX.XXXX	3A11/71

Note: Not at PCA for SLR-2000

Line 3

Section/Line Identifier	55	XX	A2
Data Identifier	A - Z	HEADER LUNR	A12

Line 4

Section/Line Identifier	56	XX	A2
Reference Time	Year	XXXX	A5
	DOY	XXX	A4
	Hour	XX	A3
	Minute	XX	A3
Reflector ID	100-104	XXX	A4/21

Line 5

Section/Line Identifier	57	XX	A2
Fraction of a Julian Date	-	XXXXX	A6
AZM Position	Deg	XXX.XXXX	A8
ELV Position	Deg	XXX.XXXX	A8
Range	nsec	XXXXXXXXX	A13
Fraction of a Julian Date	-	XX.X	
AZM Position	Deg	XXX.XXXX	A6
ELV Position	Deg	XXX.XXXX	A8
Range	nsec	XXXXXXXXX	A8
		XX.X	A13/72

Line 6

Section/Line Identifier	58	XX	A2
Fraction of a Julian Date	-	XXXXX	A6

AZM Position	Deg	XXX.XXXX	A8
ELV Position	Deg	XXX.XXXX	A8
Range	nsec	XXXXXXXXX	A13
Fraction of a Julian Date	-	XX.X	
AZM Position	Deg	XXX.XXXX	A6
ELV Position	Deg	XXX.XXXX	A8
Range	nsec	XXXXXXXXX	A8
		XX.X	A13/72

Line 7

Section/Line Identifier	59	XX	A2
Fraction of a Julian Date	-	XXXXX	A6
AZM Position	Deg	XXX.XXXX	A8
ELV Position	Deg	XXX.XXXX	A8
Range	nsec	XXXXXXXXX	A13
Fraction of a Julian Date	-	XX.X	
AZM Position	Deg	XXX.XXXX	A6
ELV Position	Deg	XXX.XXXX	A8
Range	nsec	XXXXXXXXX	A8
		XX.X	A13/72

CONTROLLER RANGE DATA FORMATS

Ranging Characteristics

	UNITS VALUE	CHARS. SCALE	FMT.
Line 0			
Section/Line Identifier	60	XX	A2
Data Identifier	A - Z	RANGE CHAR.	A12
Line 1			
Section/line Identifier	61	XX	A2
Time of Incident	DOY	XXX	A4
	SEC	XXXXX	A6
Temperature	Celsius	SXX.X	A6
Humidity	%	XXX	A4
Barometric Pressure	mbar	XXXX.XX	A8
Time Bias	ms	SXXXXX	A7
Azimuth Bias	Deg.	SXX.XXXX	A9
Elevation Bias	Deg.	SXX.XXXX	A9
Cross Track	mdeg	SXX.XX	A7
System Mode (1)	0 - 9	X	A2
Along Track	mdeg	SXX.XX	A7/71

Note: Frame output only when a parameter change occurs.

Line 2			
Section/Line Identifier	62	XX	A2
Range Window	ns	XXXXX	A6
Range Window/Window	ns	XXXXX	A6
Range Bias	ns	SXXXX.X	A8
Beam Divergence	mdeg	XX.XX	A6
N.D. Value	-	X.X	A4
Transmit Optics - In/Out	1/0	X	A2
Daylight Filter - In/Out	1/0	X	A2
PMT Voltage	V	XXXX	A5
PMT Voltage 2-color (48-inch)	V	XXXX	A5
CMOS Range Zero Time (MLRS)	sec	XXXXXXXXX	A11
HSLR Operator Override	1/0	X	A2
HSLR Status (2)	0 - 7	X	A1
Laser Interlock Unit Status (3)	0 - FF	XX	A3
Laser Fires		XXXXX	A6

Line 3

Section/Line Identifier	63	XX	A2
Spatial Filter	0.1 arcsec	XXXX	A5
Spectral Filter	0.1 Angstroms	XXXX	A5
Number of Shots(MLRS)		XXXXXX	A7
Detector(4)	0/1/2	X	A2/21

Line 4

Section/Line Identifier	64	XX	A2
Mode (5)	0/1/2	X	A2
Scan type (6)	0/1/2	X	A1
Focus		SXXX	A5
Focus Offset		XXX	A4
Laser Voltage	Volts	XXXX	A5
Laser Change Voltage	Volts/D Rep-rate	XX.XX	A5
Sky Clarity (7)	0/1/2/3	X	A2
Number of records		XXXXX	A6
Percent noise	%	XX.X	A4/36
Wind Speed	m/sec	XXX	A4
Wind Direction	degree	XXX	A4
Vaisala	Precipitation type	XXX	A4
Visibility	Km	XX	A3

CONTROLLER RANGE DATA FORMATS
Range Characteristics Reference

(1) System Mode : 0 = Standby

1 = Stow	6 = Tracking/Dump
2 = Tracking	7 = Calibration/Dump
3 = Calibration	8 = Fixed/dump
4 = Fixed	9 = Tart/Dump
5 = Tart	

(2) HSLR Status: 7 = Off/Manual/Nonamp
6 = Off/Manual/Amp
5 = Off/Computer/Nonamp
4 = Off/Computer/Amp
3 = On/Manual/Nonamp
2 = On/Manual/Amp
1 = On/Computer/Nonamp
0 = On/Computer/Amp

(3) Laser Interlock Status:
Bit 8 = 10 N.D. Block
Bit 9 = Transmit Filter
Bit 10 = Beam Block
Bit 11 = Console Panel Inhibit
Bit 12 = Laser Room Inhibit
Bit 13 = Mount Area Inhibit
Bit 14 = Stairwell Movement
Bit 15 = Radar Detect

(4) Detector:
0 Varian
1 MCP
2 SPAD

(5) Tracking Mode:
0 Acquisition
1 Tracking
2 Test

(6) Scan Mode:
0 Not scanning
1 Angle Scan
2 Range Scan
3 Angle and Range Scan

(7) Sky Clarity:
0 No data
1 Clear
2 Partly cloudy
3 Totally cloudy

CONTROLLER RANGE DATA FORMATS
Data

	UNITS VALUE	CHARS. SCALE	FMT.
Line 0			
Section/Line Identifier	70	XX	A2
Data Identifier	A - Z	RANGE DATA	A12
Line 1			
Section/Line Identifier	71	XX	A2
Data Validation (Hits/All)	1/0	X	A2
Time of Laser Fire	Year	XXXX	A5
	DOY	XXX	A4
	Sec	XXXXX	A6
	ns	XXXXXXXXX	A10
AZM Designate	Deg	XXX.XXXX	A9
ELV Designate	Deg	XXX.XXXX	A9
RNG Designate, 2-way TOF	ns	XXXXXXXXXX.	A15
Elevation Refraction	Deg	XXX	A7
Range refraction	ns	.XXXX	A7/76
		XX.XXX	
Line 2			
Section/Line Identifier	72	XX	
AZM Encoder	Deg	XXX.XXXX	
ELV Encoder	Deg	XX.XXXX	
RNG Measurement, 2-way TOF	ns	XXXXXXXXXX.	
Transmitter Delay	ns	XXX	
Transmit Energy	vns	XXXXXXXXX	
Receive Energy	vns	XXXX	
		XXXX	
AZM Mount Model Correction	Deg.	SXX.XXXX	
ELV Mount Model Correction	Deg.	SXX.XXXX	
Quadrant detected	1-4	xx	
Line 3			
Section/Line Identifier	73	XX	A2
2-Color Parameters (48-inch)			
Delta Measured, 2-way TOF	ns	XXX.XXX	A8
Predicted Delta, 2-way TOF	ns	XXX.XXX	A8
Transmit Energy	vns	XXXX	A5
Receive Energy	vns	XXXX	A5
Elevation Refraction	deg	.XXXX	A7/35
Line 4			
Section/Line Identifier	74	XX	A2
Multi-Channel (Fire) Verniers (MLRS)			
fire times	ms	XXXXXXXX.X	A11
fire verniers	counts	XXXX	8A5
delta zero time	sec	XXXXX	A6/59

Line 5			
Section/Line Identifier	75	XX	A2
Multi-Channel (Return) Verniers (MLRS)			
Return Times	ms	XXXXXXXX.X	A11
Return Verniers	Counts	XXXX	8A5/53
Line 6			
Section/Line Identifier	76	XX	A2
Time of Angle	Year	XXXX	A5
	DOY	XXX	A4
	Sec	XXXXX	A6
	Milliseconds	XXX	A4
AZM Designate	Deg	XXX.XXXX	A9
ELV Designate	Deg	SXX.XXXX	A9
AZM Encoder	Deg	XXX.XXXX	A8
ELV Encoder	Deg	SXX.XXXX	A8
AZM Mount Model	Deg	XXX.XXXX	A8
ELV Mount Model	Deg	SXX.XXXX	A8/71
Line 7			
Section/Line Identifier	77	XX	A2
AZM Scan Offset	Deg	SX.XXXX	A7
ELV Scan Offset	Deg	SX.XXXX	A7
AZM Quad Detector Angle Offset	Deg	S.XXXX	A6
ELV Quad Detector Angle Offset	Deg	S.XXXX	A6
AZM Manual Offset	Deg	SX.XXXX	A7
ELV Manual Offset	Deg	SX.XXXX	A7
ELV Refraction Correction	Deg	S.XXXX	A6
Dome AZM Encoder	Deg	XXX.XXXX	A8
Dome AZM Bias	Deg	SXX.XXXX	A8/62

CONTROLLER RANGE DATA FORMATS

Ancillary Data

	UNITS VALUE	CHARS. SCALE	FMT.
Line 0			
Section/Line Identifier	80	XX	A2
Data Identifier	A - Z	ANCILLARY DATA	A12
Line 1			
Section/Line Identifier	81	XX	A2
Wind Speed	MPH	XXX	A4
Wind Direction	NW	XX	A3
Seeing Value	.1 arcsec	XXXX	A5
Sky Coverage		XX	A3
Energy	joules(lunar)	XX	A3
Dark Count	KHz	XXX	A4
Site Background count	KHz	XXX	A4
Crew Initials			A20/48
Line 2			
Section/ line Identifier	82	XX	A2
Comments			A2
Comments			A80
Comments			A80
			A80/242
Line 3			
Section/Line Identifier	83	XX	A2
Time of Focus	Year	XXXX	A5
	DOY	XXX	A4
	Sec	XXXXX	A6
	ms	XXX	A4
MAX Travel	-	XXXX	A5
MIN Travel	-	XXXX	A5
Star Number	-	XXXX	A5
Magnitude	-	XX.X	A4
Final Image size	-	X.X	A3
Focus Offset	-	XXX.X	A5
Star Offset from AZM Center	Deg	XXX.XX	A6
Star Offset from ELV Center	Deg	SXX.X	A5
Temperature	Degrees C	XXX.X	A5
Dew Point	Degrees C	XXX.X	A5
Sky Clarity	0/ 1/ 2	X	A2/70

Daily Diary

Date 09/02/97
Column Headers (do we include)
sic %5c
Start Time hhmm %3d%2d
End Time hhmm %3d%2d
Number of fires %6d
Number of returns %6d
Pass RMS in millimeters %4d
Average Pressure millibars %5.1f
Average Temperature degrees C %4.1f
Average Humidity %3d
Cloud cover C P M T %1c
Signal to Noise Ratio %4.2f
Applied system delay(nano seconds) %5.2f
Ground visibility C R F %1c

STAR %5c
Start Time hhmm %3d%2d
End time hhmm %3d%2d
Calibration rms arc seconds %3.1f
Average Pressure millibars %5.1f
Average Temperature degrees C %4.1f
Average Humidity %3d
Cloud cover C P M T %1c
Ground visibility C R F %1c
offsets from last calibration # of coefficients %2.5

FOCS %5c
Start Time hhmm %3d%2d
New value %4d
Temperature degrees C %4.1f
Star number %4d

Conditions orange or greater
Start Time hhmm %3d%2d
Duration hhmm %3d%2d
Condition level %1c
Reason %60c

E-mail Distribution File Format

One E-mail address per line

Emergency Contact File

Number of numbers %2d
Local country code %4d

Contact phone number lines
Phone number including country code %13d
Contacts name(not used by the computer) %60c

Gravity Model Format

header line %20c

date lines
RECOEF %6c
N %2d
M %2d
C %15.8le
S %15.8le

Maintenance file format

maintenance type number (on bit per test)	%4d
Duration in minutes	%2d
Maximum level that this function can be performed in	%1d
ASCII description	%60c

Error File Format

(Capturing errors from the Satellite and Calibration Processors)

Record 1:

Description	Example
Data set file name	logx_y1997d252t1201
Routine name where the error occurred	Open_Controller_File

Record 2:

Description	Example
Message	"error opening satellite file"

FITS Standard for SLR2000 Camera Image Files

Recognizing the need to adopt a header and data file format for SLR2000 camera images, several industry standards including BMP, PCX, TIF, and GIF were considered, along with FITS (Flexible Image Transport System). FITS is recommended because it is approved by the NASA/Science Office of Standards and Technology, it is well supported and widely used at scientific centers around the world, and writing code for it is easy.

The FITS header format for SLR2000's two-dimensional data is straight forward. Only these five keywords are required: SIMPLE, BITPIX, NAXIS, NAXISn, and END. As an example, the star camera data can be headed as follows:

```
SIMPLE  =                     T
BITPIX  =                     8
NAXIS   =                     2
NAXIS1  =                   242
NAXIS2  =                   242
END
```

The SIMPLE keyword indicates whether the file conforms to this standard. BITPIX is the number of bits in a data value. NAXIS is the number of axes (dimensions) in the array. NAXISn are the key words representing the number of data points along axis 'n'. END marks the end of the header section. Each keyword record is an 80 byte card image, and may include a comment following a '/' delimiter. Blanks are used to fill out individual records.

Many other keywords are available which may pertain to the observation, the source of the data, or additional comments. The header must consist of 36 card images, with blanks filling empty ones.

The data part of the file consists of a byte stream with no blank spaces or fill. Thus, for star camera images this portion of the file will consist of 242 * 242 = 58,564 bytes. I recommend that NAXIS1 should refer to samples (columns) and NAXIS2 refer to lines (rows). Then the data area will consist of samples 1 to 242 of line 1, followed by samples 1 to 242 of line 2, etc. This can be indicated unambiguously by optional keywords if desired.

Additional details about the FITS standard are available on-line through the FITS Support Office home page at http://ssdoo.gsfc.nasa.gov/astro/fits/fits_home.html.

IV File Format

Header line	
SIC	%4d
Prediction version number	%3d
Pole position X Milli-arcsecond	%4d
Pole position Y Milli-arcsecond	%4d
Greenwich hour angle	%12.10lf
Day of year	%3d
Year	%4d
Data lines	
Modified Julian date	%16.10lf
Satellite X position(meters)	%16..6lf
Satellite Y position(meters)	%16..6lf
Satellite Z position(meters)	%16..6lf
Satellite X velocity(meters/sec)	%16..9lf
Satellite Y velocity(meters/sec)	%16..9lf
Satellite Z velocity(meters/sec)	%16..9lf
Satellite X acceleration(meters/sec/sec)	%16..13lf
Satellite Y acceleration(meters/sec/sec)	%16..13lf
Satellite Z acceleration(meters/sec/sec)	%16..13lf

Mask File format

Line 1	
upper mask array size Mmask	%2d
Lower mask array size Lmask	%2d
Line 2 to Mmask + 1	
Azimuth	%7.3f
Elevation	%6.3f
Line Lmask + 2 to Lmask + 2	
Azimuth	%7.3f
Elevation	%6.3f
Number of Time mask elements Tmask	%2d
Mask line	
day of week number	%1d
Start Hour	%2d
:	%1c
start Minute	%2d
End Hour	%2d
:	%1c
End minute	%2d
Level	%1d

Mask File Format (Continues)

The system mask consists of two submasks:
1) the mount travel mask (mmask), and
2) the laser fire mask (lmask).

The mount travel mask places a limit on the actual travel of the mount. The laser fire mask gives the region where the laser can fire. The masks are given in AZ/EL, in degrees. The mask arrays must be monotonically increasing in AZ. Each submask consists of two arrays: an upper limit array and a lower limit array. For the laser fire mask, if the upper limit and the lower limit are equal, then no lasing can occur at that azimuth for any elevation.

For both mmask and lmask, the upper & lower arrays are as follows:
There are NMx points in array#x (x=1,2).
If NMx = 0 ==> there is no limit.
If NMx = 1 ==> there is a constant EL limit for all AZ.
If NMx > 1 ==> straight line interpolation between conseq pts

While the mount travel mask is fixed, the laser fire mask can be time dependent. Up to 14 laser fire masks can be accommodated for the week. For each fire mask, there is an associated time period. If periods during the day do not have an associated laser mask, then it is assumed that no lasing can occur during those periods . The time period is specified by a 0-7 for Sunday through Saturday, with GMT time to the nearest minute (HH:MM)...

Laser mask #	START	STOP
1	D/HH:MM	D/HH:MM
2	etc	

If lasing can occur all the time during the entire week, then the laser fire mask time period would be specified by START=0/00:00 and STOP=7/24:00.

There must be both a mount mask and a laser mask. If there are no limits whatever on mount travel or lasing, then both the mount mask and the laser mask would have NM1=NM2=0, and the time period would be specified as explained above.

MERIT II FORMAT REVISION 2

A 130 byte ASCII character record
13,000 fixed byte blocks

Field	Description	Example
1-7	Satellite COSPAR ID - 7-digit COSPAR satellite identification number	‘7603901’
8-9	Year of Century - 2 digits with leading zero fill	‘87’
10-12	Day of Year - 3 digits with leading zero fill	‘76’
13-24	Time of Day - from midnight GMT with a 0/.1 microsecond granularity and leading blank fill	‘ 36005000000’
25-28	Station ID - 4-digit monument identification number from the NASA Directory of Station Locations	‘7105’
29-30	Crystal Dynamics Project: System Number - 2-digit system number assigned by the Crustal Dynamics Project with a leading zero fill	‘07’
31-32	Crystal Dynamics Project: Occupancy Sequence Number - 2-digit monument occupancy number assigned by the Crustal Dynamics Project with a leading blank fill	‘02’
33-39	Azimuth - the geocentric or true azimuth angle with a 0.1 millidegree granularity and leading blank fill	‘ 987500’
40-45	Elevation - the geocentric or true elevation angle with a 0.1 millidegree granularity and leading blank fill	‘292500’
46-57	Laser Range - in units of two-way time with a 1 picosecond granularity and leading blank fill	‘ 52035998000’
58-64	Laser Range Standard Deviation - in units of two-way time with 1 picosecond granularity and leading blank fill	‘ 66’
65-68	Wavelength - 0.1 nanometer granularity with leading blank fill	‘5320’
69-73	Surface Pressure - 0.1 millibar granularity with leading blank fill	‘10135’
74-77	Surface Temperature - 0.1 degree granularity with leading blank fill	‘2905’
78-80	Relaltive Hunmidity at surface - percentage with leading blank fill	‘ 55’
81-85	Tropospheric refraction correction - a round-trip correction with 1 picosecond granularity and leading blank fill	‘33956’
86-91	Centrer of Mass Correction - a round-trip correction with 1 picosecond granularity and leading blank fill	‘ 1601’
92-96	Receive Amplitude - a positive linear scale value, usually between 0 and 2000, with leading blank fill	‘ 700’

Field	Description	Example
97-104	Applied System Delay - the two-way system delay applied in the current record with a 1 picosecond granularity and leading blank fill	‘ 95942’
105-110	Calibration Delay Shift - a measure of two-way calibration stability with a 1 picosecond granularity and leading blank fill. The type of shift is indicated in column 127.	‘ 33’
111-114	Calibration Standard Deviation - in units of two-way time with a 1 picosecond granularity and leading blank fill	‘ 40’
115	Normal Point Window Indicator - Indicates whether of not the record represents a normal point and the time span of the normal point: 0 = Range not a normal point 2 = 10-second normal point 3 = 15-second normal point (TOPEX) 4 = 20-second normal point 5 = 30-second normal popint (low orbit only) 6 = 1-minute normal point (historical data only) 7 = 2-minute normal point (LAGEOS) 8 = 3-minute normal point (historical data only) 9 = 5-minute normal point (ETALON)	‘0’
116-119	Number of raw ranges compressed into normal point - leading blank fill	‘ ‘
120	Epoch event - indicates the time event reference. Currently, only 1 and 2 are used for laser data. 0 = Ground receive time 1 = Satellite transmit time (standard for LAGEOS) 2 = Ground transmit time 3 = Satellite receive time	‘1’
121	Epoch Time Scale - indicates the time scale reference. Other values may appear for historical data. 3 = UTC (USNO) 7 = UTC (BIH)	‘3’
122	Angle Origin Indicator - source of angle values. 0 = Unknown (Converted from MERIT I) 1 = Computed (from range) 2 = Command (predicts and operator inputs) 3 = Measured (calibrated instrument readings)	‘3’
123	Tropospheric Refraction Correction Indicator - Range 0 = Data has been corrected using the Marini-Murray formula. 1 = Data has not been corrected.	‘0’

Field	Description	Example
124	Center of Mass Correction Application Indicator 0 = Applied 1 = Not applied	‘0’
125	Receive Amplitude Correction Indicator 0 = Data has been receive-amplitude corrected 1 = Data has not been receive-amplitude corrected	‘1’
126	System Calibration Method Indicator 0 = External Calibration 1 = Internal Calibration 2 = Burst Calibration - 3-minute bursts 3 = Override Calibration	‘0’
127	Calibration Delay Shift Indicator - specifies the type of delay shift represented in columns 105 to 110. 0 = Pre to Post shift 1 = Peak to Peak shift	‘0’
128	System Configuration Indicator - is a flag to be incremented for every major system configuration change. The flag will be initially set at ‘1’ at the time of MERIT II implementation. Data prior to MERIT II that is converted to MERIT II format will have a flag of ‘0.’	‘1’
129	Format Revision Number Indicator - Indicates the version of the MERIT II format for the current record. Data prior to MERIT II that is converted to MERIT II format will have a flag of ‘0.’	‘2’
130	Release Flag Indicator - indicates when this record first appeared on a release tape. Foreign stations which send release tapes to the DIS will use a numbering scheme beginning with ‘1.’ Release tapes from the DIS will have a labeling scheme beginning with the letter ‘A.’ Non-operational engineering data will have a release flag of ‘Z.’ Data released prior to the MERIT II implementation will have a release flag of ‘0.’	‘A’

ON-SITE NORMAL POINT FORMAT

Header Record

Column	Description	Example
1-7	Satellite identifier - 7-digit COSPAR identification number	'7603901'
8-9	Year of century	'89'
10-12	Day of Year	'079'
13-16	Crustal Dynamics Project Pad ID - 4-digit monument identification	'7105'
17-18	Crustal Dynamics Project - 2-digit system number	'07'
19-20	Crustal Dynamics Project - 2-digit occupancy sequence number	'02'
21-24	Wavelength of the laser The user of the data should interpret the value given as follows: 3000 - 9999: units are 0.1 nanometer 0001 - 2999: units are 1.0 nanometer For the station generating the data, the rule is: Wavelength in rate 0.3000 - 0.9999 microns: unit 0.1 nanometer Wavelength in rate 1.000 - 2.999 microns: unit 1.0 nanometer	'5321'
25-32	Calibration system delay (two-way value in picoseconds)	'00095942'
33-38	Calibration delay shift (two-way value in picoseconds)	'000033'
39-42	Root Mean Square (RMS) of raw system delay values from the mean. Two-way value in picoseconds. If pre-and post-pass calibrations are made, use the mean of the two RMS values, or the RMS of the combined data set.	'0040'
43	Normal point indicator (an integer from 0 to 9) 0 = not a normal point 1 = 5-second normal point (GFZ-1) 2 = LLR normal point 3 = 15-second normal point (TOPEX) 4 = 20-second normal point 5 = 30-second normal point 6 = 1-minute normal point 7 = 2-minute normal point (LAGEOS) 8 = 3-minute normal point 9 = 5-minute normal point	'7'
44	Epoch time scale indicator 3 = UTC (USN) 4 = UTC (GPS) 7 = UTC (BIPM) (BIH prior to 1988)	'3'

Header Record

Column	Description				Example
45	System calibration method and delay shift indicator. Indicates the type of calibration and the type of calibration shift given in columns 33 - 38.				
		Pre- to Post-Pass Calibration Shift	Minimum to Maximum Calibration Shift		
	External Cal Internal Cal Burst Cal Some other cal Not used	0 1 2 3 4	5 6 7 8 9		'0'
46	System Change indicator (SCH). A flag to increment for every major change to the system (hardware or software). After the value '9' return to '0,' and then continue incrementing. The station and data centers should keep a log in a standard format of the value used, the date of the change, and a description of the change.				'0'
47	System Configuration Indicator (SCI). A flag used to indicate alternative modes of operation for a system (e.g., choice of alternative timers or detectors, use of a different mode of operation for high satellites). Each value of the flag indicates a particular configuration, which is described in a log file held at the station and at the data centers. If only a single configuration is used then use a fixed value. If a new configuration is introduced then use the next higher flag value. If the value exceeds '9' then return to '0,' overwriting the previous configuration flag (it is not likely that a station will have 10 current possible configurations).				'1'
48-51	Pass RMS from the mean of raw range values minus the trend function, for accepted ranges (two-way value in picoseconds).				'0065'
52	Data quality assessment indicator For LLR data: 0 = Undefined or no comment 1 = Clear, easily filtered data, with little or no noise 2 = Clear data with some noise, filtering is slightly compromised by noise level. 3 = Clear data with a significant amount of noise, or weak data with little noise. Data are certainly present, but filtering is difficult 4 =Unclear data; data appear marginally to be present, but are very difficult to separate from noise during filtering. Signal to noise ration can be less than 1:1. 5 = No data apparent				'0'
53-54	Checksum - an integer value equal to the sum of the integers in columns 1-52, modulo 11 (optional)				'54'
55	Format revision number indicator. Value '1' for this 1997 revision. Implied value '0' or 'space' for original 1990 release.				'1'

Data Record

Column	Description	Example
1-12	Time if day of laser firing, from 0 hours UTC in units of microseconds. Value is given module 86400 if pass crosses 24 hours UTC.	‘214360786545’
13-24	Time of flight corrected for system delay, in picoseconds. Not corrected for atmospheric delay, nor to the satellite center-of-mass.	‘052035998000’
25-31	Bin RMS from the mean of the raw range values minus the trend function, for accepted ranges. Two-way value in picoseconds. If point is a single raw data point, then use pass RMS.	‘’0000066’
32-36	Surface pressure, in units of 0.1 millibar	‘10052’
37-40	Surface temperature in units of 0.1 degree Kelvin	‘2932’
41-43	Relative humidity at surface in percent.	‘092’
44-47	Number of raw ranges (after editing)	‘0108’
48	A flag to indicate the data release: 0 = first release of data 1 = first replacement release of the data 2 = second replacement release, etc.	‘0’
49	For SLR data: not used For LLR data: integer seconds of two-way time of flight (columns 13-24 contain the fractional part.)	‘2’
50	For SLR data: not used For LLR data: normal point window indicator. Indicates time span of the normal point (can be variable from point to point). 1 = <5 minutes 2 = 1 minute 3 = 15 minutes … 9 = >50 minutes	‘1’
51-52	For SLR data: not used For LLR data: signal to noise ratio, in units of 0.1, e.g., 00 = No information 01 = Signal/noise = 0.1 … 99 = Signal/noise = 9.9 or greater	‘00’
53-54	Checksum - integer value = to the number of sums in digits 1-52, modulo inn (optional)	‘62’

Quicklook Format (Header Record)

COLUMN	DESCRIPTION	EXAMPLE
1-7	Satellite - 7-digit COSPAR identification number	'7603901'
8-9	Year of century	'89'
10-12	Day of year	'079'
13-16	Crustal Dynamics Project Pad ID - 4-digit monument identification	'7105'
17-18	Crustal Dynamics Project 2-digit system number	'07'
19-20	Crustal Dynamics Project 2-digit occupancy sequence number	'02'
21-24	Wavelength of laser (units of 0.1 or 1.0 nanometer, depending on wavelength value) 0001 - 1999 (1.0 nanometer) 3000 - 9999 (0.1 nanometer)	'5321'
25-32	Calibration system delay (two-way value in picoseconds)	'00095942'
33-38	Calibration delay shift (two-way value in picoseconds)	'000033'
39-42	Root Mean Square (RMS) of system delay value (two-way value in picoseconds)	'0040'
43	Normal point window indicator: 0 = not a normal point 1 = 5-second normal point (GFZ-1) 2 = 10-second normal point 3 = 15-second normal point (TOPEX) 4 = 20-second normal point 5 = 30-second normal point 6 = 1-minute normal point 7 = 2-minute normal point (LAGEOS) 8 = 3-minute normal point 9 = 5-minute normal point (ETALON)	'7'
44	Epoch time scale indicator 3 = UTC (USNO) 4 = UTC (GPS) 7 = UTC (BIPM) (BIH prior to 1988)	'3'
45	System calibration method indicator 0 = external calibration 1 = internal calibration 2 = burst calibration 3 = override calibration	'0'
46	Calibration delay shift indicator - specifies delay shift type in columns 33 - 38. 0 = pre to post shift 1 = minimum to maximum shift	'0'
47	System configuration flag indicator	'1'
47-51	Pass RMS of accepted ranges (two-way value in picoseconds	'0065'
52	unused (zero-filled)	'0'
53-54	Checksum - an integer value equal to the sum of the integers in columns 1- 52, modulo 100	'54'

Sampled Engineering Data Record

COLUMN	DESCRIPTION	EXAMPLE
1-12	Time of day of laser firing, from 0 hours UTC, in units of 0.1 microsecond	'214360786545'
13-24	Two-way time of flight with no corrections applied, in picoseconds	'052035998000'
25-29	Surface pressure, in units of 0.1 millibar	'10052'
30-33	Surface temperature, in units of 0.1 degree Kelvin	'2932'
34-36	Relative humidity at surface, in percent	'092'
37-44	Internal burst calibration system delay	'00003124'
45-48	Relative signal strength for the return (units of measure determined by individual stations)	'0789'
49	Angle origin indicator - source of angle values in columns 50 - 62: 0 = unknown 1 = computer (from range) 2 = command angles - predicted angles with refraction correction and crew bias 3 - measured angles - encoder readings with mount model corrections removed to give actual azimuth and elevation as affected by refraction	'3'
50-56	Azimuth angle in units of 0.0001 degree, using local reference system (North = 0, East = 90)	'0981501'
57-62	Elevation angle in units of 0.0001 degree, using local reference system (zenith = 90)	'292501'
63-67	Unused (zero-filled)	'00000'
68-69	Checksum - an integer value equal to the sum of the digits in columns 147, module 100	'05'

Satellite File Format

Column	Description	Type	Example
1-9	Satellite Name	Character *9	LAGEOS
10	Blank		
11-14	Satellite Identification Code *	Integer *4 (14)	1155
15	Blank		
16-22	Satellite COSPAR Identification Number	Integer *4 (17)	7603901
23	Blank		
24-30	Constant Center of Mass in meters	Real *8 (F7.4)	00.2510
31	Blank		
32-39	Minimum Range for acceptance of satellite data in 2-way time units of seconds	Real *8 (F8.6)	0.033300
40	Blank		
41-48	Maximum Range for acceptance of satellite data in 2-way time units of seconds	Real *8 (F8.6)	0.066700
49	Blank		
50	Start degree for Least Squares fit	Integer *4 (I1)	3
51	Blank		
52-55	Normal point bin size in seconds	Integer *4 (I5)	00120
56	Blank		
57	Additional fit Flag (Additional fit necessary for low satellites)	Logical *1 (L1)	f
58	Blank		
59-60	Gravity Model Indicator	Integer *4 (I2)	08
61	Blank		
62-65	Integrator step size in seconds	Integer *4 (I4)	0360
66	Blank		
67-68	Pulses per Second	Integer *4 (I2)	05
69	Blank		
70-77	Satellite area in meters squared	Real *8 (F8.5)	01.13000
78	Blank		
79-84	Satellite Mass in kilograms	Real *8 (F6.1)	0411.0
85	Blank		
86-90	Satellite Reflectivity	Real *8 (F5.3)	1.000
91	Blank		
92-93	Unit Number for Tuned IRV File	Integer *2 (I2)	30
94	Blank		
95-102	Tuned IRV File Name (Not including suffix)	Character *8 (A8)	TUNE_LAG
103	Blank		
104-113	Lidar cross section	Real*8(F10.5)	7.0

	(in millions of meters square)		
114	Blank		
115-121	Histogram bins size (in nanoseconds)	Real*8(F7.2)	0.5
122	Blank		
123-129	Frame length Seconds	Real*8(F7.3)	16.5
130	Blank		
131-133	Super Frame length (Three frames constitutes a super frame)	Integer*4(I3)	3
134	Blank		
135-137	Number of valid frames needed in superframe (2 of 3 frames must contain valid data for algorithm to determine that data contains signal)	Integer*4(I3)	2
138	Blank		
139-133	Acquisition elevation lower limit (in degrees)	Real*8(F5.2)	20.20
134	Blank		
135-141	Blanking time(in microseconds)	Real*8(F7.3)	100.0
142	Blank		
143-150	Acquisition range window(day) (in nanoseconds)	Real*8(F8.3)	100.0
151	Blank		
152-159	Acquisition range window(night (in nanoseconds)	Real*8(F8.3)	1000.0
160	Blank		
161-168	Limit on range scan (in microseconds)	Real*8(F8.3)	0.5
169	Blank		
170-179	Limit on angular scan (in arcseconds)	Real*8(F10.1)	30.0
180	Blank		
181-188	Minimum range window	Real*8(F8.3)	100.0

* Negative satellite identification codes are for ground targets

Schedule Input File Format

First line contains :

Length of calibration %2d
Calibration frequency (minutes) %3d

One line per satellite :

Satellite name %8c
SIC %4d
activate flag 1=support 0 = do not schedule %1d
priority 0 = highest priority
minimum useful tracking time in minutes
maximum pass length
lighting conditions 1= no day time tracking 2= no night time tracking 0 = no restrictions %1d
Minimum tracking elevation %2d
Vector file name %3c

Schedule Output File

SIC 9999 = calibration 9997 = maintenance 9990 = dome close 9991= no laser star cal only 9992 ground cal only %4d
date yymmdd %6d
day of year %3d
event start time hh:mm %2d:%2d
event duration hh:mm %2d:%2d
event priority %2d
event azimuth %3d
event elevation %3d
satellite prediction file %3c

System Change Indicator

Byte one is to be incremented for each major hardware
Byte two is to be incremented for each software change

Station Information Database (SIDB) Format

LINE 0

Description	Type	Example
Station Identification Code	Character*3	87
End Year	Character*5	1986
End Day of Year	Character*4	271
Start Year	Character*5	1980
Start Day of Year	Character*4	1
Station Name	Character*7	m10704
Marker ID:	Character*5	7105

Monument Identification Number

Crustal Dynamic Project System Number	Character*3	
Crustal Dynamic Project		
Occupancy Sequence Number	Character*3	01
Configuration Flag	Character*3	1
Latitude(degrees)	f11.6	39.020540
Longitude (degrees)	f11.6	283.172000
Height(meters)	f8.3	53.053

LINE 1

Description	Type	Example
Wavelength 1 (angstroms)	f6.0	5320.
Wavelength 2 (angstroms)	f6.0	3553.
Azimuth Offset (degrees)	f11.6	-23.992300
Elevation Offset (degrees)	f11.6	-29.100000
Barometric Pressure Offset (millibar)	f4.1	0000.0
System Delay Time Offset (nano seconds)	f8.4	0.0000
Transmit Delay Time (micro seconds)	f6.1	8400.0

LINE 2

Description	Type	Example
Beam Divergence (micro radians)	f7.3	0000000.000
Azimuth Stow Position (degree)	f5.1	90.0
Elevation Stow Position (degrees)	f4.1	12.0
Transmit Optics Correction (millimeters)	f3.1	000.0
Daylight filter (millimeters)	f4.1	10.1
Spectral filter width (angstrums)	F4.1	xx.x

†The Station Information Database is an ASCII file.
The key field is columns 1-9. The most recent version of the database is located on DAN in: /data/lib/geod_sxx....

Summary Database Capture File Format (.PCR)

FIELD	BYTES	DESCRIPTION	FORMAT
1	01-17	Station-Date-Time	A17
2	18-20	Occupation Number	A3
3	21-25	Satellite ID Code	I5
4	26-28	Calibration Target Indicator	A3
5	29-37	Target Distance (mm)	F9.0
6	38-43	Applied Cal Observations	I6
7	44-49	Applied Cal Rejects	I6
8	50-54	Applied Cam RMA (mm)	F5.1
9	55-62	Applied Delay (mm)	F8.1
10	63-69	Pre to Post Cal Shift (mm)	F7.2
11	70-75	Satellite Observations	I6
12	76-81	CDPLP Fit Rejects	I6
13	82-89	CDPLP Fir RME (mm)	F8.1
14	90-95	Generic NP System IRV Fit Reject	I6
15	96-103	Generic NP System IRV Fit RMS (mm)	F8.1
16	104-111	DDNA RMS (mm)	F8.1
17	112-116	Satellite A1 Mean	F5.2
18	117-121	Satellite A1 RMS	F5.2
19	122-126	Satellite PMT Mean (v)	I5
20	127-131	Satellite PMT RMS (v)	F5.2
21	132-135	Satellite Transmit Mean	I4
232	136-142	Satellite Transmit RMS	F7.2
23	143-146	Satellite Receive Mean	I4
24	147-153	Satellite Receive RMA	F7.2
25	154-162	Satellite Transmit Delay Mean (ms)	F9.2
26	163-172	Satellite Transmit Delay RMS (ms)	F9.2
27	173-180	Time Correction (ms)	F8.5
28	181-186	Range Correction (mm)	F6.1
29	187-192	ND Filter Correction (mm)	F6.1
30	193-198	Minimum Temperature (0.01°C)	I6

FIELD	BYTES	DESCRIPTION	FORMAT
31	199-204	Maximum Temperature (0.01°C)	I6
32	205-210	Mean Temperature (0.01°C)	I6
33	211-216	Minimum Pressure (0.1 mm)	I6
34	217-222	Maximum Pressure (0.1 mm)	I6
35	223-228	Mean Pressure (0.1 mm)	I6
36	229-232	Humidity (%)	I4
37	233-234	Meteorological Override Flag	A2
38	235-242	Latitude (d)	F8.2
39	243-250	Height (m)	F8.2
40	251-252	Geodetic Override Flag	A2
41	253-254	Type of Satellite Analysis	A2
42	255-262	IRV Time Bias in CDPLP (s)	F8.2
43	263-272	IRV Range Bias in CDPLP (m)	F10.3
44	273-275	Calibration Sigma Multiplier (0.1)	I3
45	276-278	Satellite Sigma Multiplier (0.1)	I3
46	279-280	Processor Version	A2
47	281-283	Addition Number	I3
48	284-289	Processor Run Date	I6
49	290-297	Longitude)d)	F8.2
50	298-299	Target Override Flag	A2
51	300-303	Number of Normal Point Bins	I4
52	304-307	Number of Normal Point Bins wit Data	I4
53	308-311	Number of Normal Points	I4
54	312-313	Fit Source Flag A = CDPLP B = Generic Normal Point System	A2
55	314-319	Pre Calibration Skew	F6.2
56	320-327	Pre Calibration Kurtosis	F6.2
57	328-333	Post Calibration Skew	F6.2
58	334-339	Post Calibration Kurtosis	F6.2
59	340-345	Combined Calibration Skew	F6.2
60	346-351	Combined Calibration Kurtosis	F6.2

FIELD	BYTES	DESCRIPTION	FORMAT
61	352-359	Satellite Skew	F8.4
62	360-367	Satellite Kurtosis	F8.4
63	368-375	Calibration Peak	F8.4
64	376-383	Satellite Peak	F8.4
65	384-385	Day Night Indicator D = Day N = Night U = Unknown	A2
66	386-387	CDPLP, Fig Flag I = IRV fit statistics P = First polynomial fit statistics U = No fit statistics present	A2

Target Database Format

Column	Description	Type	Example
1-2	*Station Identification Code	Character *2	64
3	Target Indicator	I *2	-1
4-7	End Year	Character *4	1986
8-10	End Day of Year	Character *3	271
11-18	Target Azimuth in degrees	Real *8	24.0560
19-26	Target Elevation In degrees	Real *8	0.1570
27-34	Target Range in meters	Real *8	128.2560

The target database is a binary indexed file. The key field is columns 1-10. The most recent version of the database is located in:

DTBS$DISK[DSGDTBS.SIDB]STATION_TARGETS.DBS

* Negative satellite identification codes re for ground targets

Tuned IRV Format

General Form

Line	Contents
1	IRSTUT
2	yyyy mmm dd hh ss.s xxxxxxxxxxxx . xxxxxx yyy … zzz …
3	satt ep nno llllllll.lllllllll mmm … nnn …
4	jxpole jypole ddrate bbbb.b ccc … ddd …

Explanation

Above, lowercase letters represent variables; UPPERCASE letters are literals. Given in parentheses below is the FORTRAN format used for reading each variable.

Line	Variable	Description
1	IRSTUT	Literal string identifying start of TIV format (A80)
2	yyyy	year (I4)
2	mm	month of the year (I2)
2	dd	day of the month (I2)
2	hh	hour of the day, Greenwich Mean Time (GMT) (I2)
2	ss.s	seconds (F4.1)
2	xxx …	x-position component, in meters F18.6)
2	yyy …	y-position component, in meters F18.6)
2	zzz …	z-position component, in meters F18.6)
3	satt	satellite identification number; e.g. LAGEOS-1 = 1155 (I4)
3	ep	ephemeris identification number (I2)
3	nno	TIV sequence number (I3)
3	lll …	x-velocity component, in meters/second (F18.9)
3	mmm …	y-velocity component, in meters/second (F18.9)
3	nnn …	z-velocity component, in meters/second (F18.9)
4	jxpole	x-polar motion component, in milli-arcseconds (I6)
4	jypole	y-polar motion component, in milli-arcseconds (I6)
4	ddrate	change in Earth rotation rate, in E^{-14} radians/second
4	bbb …	checksum of all numerical TIV values, excluding position and velocity components (F6.1)
4	ccc …	checksum of position components (F18.6)
4	ddd …	checksum of velocity components (F18.9)

Star Data File Format

Header
Filename
Starting Coefficient Vector and Covariance Matrix
UT1-UTC
Temperature, Pressure, Humidity
Measurement error, rejection scale factor, Covariance scale factor

Data One of following for each star.
CATALOG#... I5
STAR#... I3
STAR MAGNITUDE..................................... F4.1
SKY SECTION.. I2
RIGHT ASCENSION (hours, minutes, seconds)............I2,I3,F7.3
DECLINATION (degrees, minutes, seconds)..............I3,I3,F6.2
YEAR... I4
MONTH AND DAY...................................... I2,I2
DAY OF YEAR.. I3
TIME OF DAY: UTC (HOUR, MINUTE, SECOND)..............I2,I2,F5.2
RAW AZ (1ST ANGLE) ENCODER (Plus offset)........... F8.4
RAW EL (2ND ANGLE) ENCODER (Plus offset)........... F8.4
AZ (1ST ANGLE) DELTA
(Bias entered + mount model correction)......... F8.4
EL (2ND ANGLE) DELTA
(Bias entered + mount model correction)......... F8.4
AZ (1ST ANGLE) ACTUAL BIAS ENTERED................. F8.4
EL (2ND ANGLE) ACTUAL BIAS ENTERED................. F8.4
Note: RA/DEC are apparent, current time.

UTC Correction File Format

Header line
Rest of lines contain following information per line:
- Date (Year, Month, Day)
- Modified Julian Day
- X and Y polar motion
- UT1-UTC (in seconds)

The UT1-UTC gives the correction to UTC for midnight on the given date. The software interpolates between the midnight values for the current time of day.

Mount Model Format

# of terms in mount model: N	
Coefficient:	
1st angle, 1st term (degrees)	F12.8
1st angle, 2nd term (degrees)	F12.8
....	
1st angle, nth term (degrees)	F12.8
2nd angle, 1st term (degrees)	F12.8
2nd angle, 2nd term (degrees)	F12.8
....	
2nd angle, mth term (degrees)	F12.8

Covariances:
Upper triangular part only.
One covariance file for each coefficient file.

Format:

1st row elements: (P(1,I),I=1,N)...................N(1X,F12.8)
2nd row elements: (P(2,I),I=2,N)...............(N-1)(1X,F12.8)
....
Jth row elements: (P(J,I),I=J,N).............(N-J+1)(1X,F12.8)
....
Nth row elements: (P(N,N))..........................(1X,F12.8**)**

# of total terms (n+m terms), Starcal RMS I3,3X,F9.6

Star Summary (Analysis) File Format

Header
Title
System name
Catalog name (eg FK5)
Fit method (Global or Kalman)
Data filename (includes date/time of starcal)
Date (Month, Day, Year) of analysis
Measurement error, number of iterations, rejection scale factor
Temperature, Pressure, Humidity

Initial mount model coefficients and standard deviations.

Data
One record per star recording...
Star Number
Catalog Number
Sky section
Time (HH:MM:SS)
Angle #1 encoder: α_1
Angle #1 Delta (Command - Encoder): $\Delta\alpha_1$
Angle #1 Bias entered
Angle #2: α_2
Angle #2 Delta (Command - Encoder): $\Delta\alpha_2$
Angle #2 Bias entered
Angle #1 residual from fit: $D1 = [\Delta\alpha_1 - \alpha_1(\text{fit})]*\text{COS}(\alpha_2)$
Angle #2 residual from fit: $D2 = \Delta\alpha_2 - \alpha_2(\text{fit})$
Residual arc: R = SQRT(D1*D1 + D2*D2)

Note: if the coordinate system is AZ/EL, then D1 = (....)*COS(EL).
Rejected stars are indicated by having a negative star number.

Summary

μ,σ of D1, D2, and R are calculated for the entire data set (excluding rejected stars) and displayed in both degrees and arcseconds.
Newly generated mount model coefficients [X] and standard deviations [s]:

X(1)	s(1,1)
X(2)	s(2,2)
....	
X(J)	s(J,J)
....	
X(N)	s(N,N)

Weather File Format

Byte	Contents	Byte	Contents
----	--------	----	--------
0	Month	50	Precip Intensity (-, ,+)
1	"	51	
2	Day	52	Visibility Range (Km)
3	"	53	"
4	Year	54	
5	"	55	DRD12 Potential (V)
6	"	56	" (decimal point)
7	"	57	"
8		58	"
9	Hour	59	"
10	"	60	
11	:	61	Precip Intensity (1-4)
12	Minute	62	
13	"	63	Clear (pct)
14	:	64	"
15	Second	65	"
16	"	66	
17		67	Haze (pct)
18	Temperature (C)	68	"
19	"	69	"
20	"	70	
21	" (decimal point)	71	Cloud (pct)
22	"	72	"
23	"	73	"
24		74	
25	Pressure (mBar)	75	Sky Code North (0-3)
26	"	76	
27	"	77	Sky Code East (0-3)
28	"	78	
29	" (decimal point)	79	Sky Code South (0-3)
30	"	80	
31	"	81	Sky Code West (0-3)
32		82	
33	Relative Humidity (%)	83	
34	"	84	
35	"	85	
36	" (decimal point)	86	
37	"	87	
38		88	
39	Wind Speed (m/s)	89	
40	"	90	
41	"	91	
42		92	
43	Wind Azimuth (Deg)	93	
44	"	94	
45	"	95	
46		96	
47		97	\r (for serial only)
48	Precipitation Code	98	\n (for serial only)
49	"	99	\0 (for serial only)
		99	\n (for file only

Star Default File (star_default.dat)

Line 1: Title and Date
/* Star Calibration Default for 48Inch: 6/17/93 */

Line 2: System name
/* 48 Inch Telescope */

Line 3: Maximum number of stars that table should hold (=< 200)
200

Line 4: Initial star selection method (when program first starts)
BRIGHT

Line 5,6: Max/Min star magnitude for BRIGHT star selections
0.0
2.5

Line 7,8: Max/Min star magnitude for GRID star selections
3.0
3.5

Line 9: Alignment star (POLARIS or other) FK5 catalog number
907

Line 10: Number of stars for Satellite Arc Selection
20

Line 11: Estimated measurement error (in degrees: 1 sigma)
0.0003

Line 12: Maximum number of iterations of star rejection loop
20

Line 13: Multiplier of starcal RMS for star rejection
2.0

Line 14: Multiplier for scaling the covariance matrix
3.0

Line 15: Option for the covariance matrix (1=scale, 2=initialize)
2

Line 16: The initialized covariance matrix (diagonal elements)
10.,10.,10.,....................10.

Line 17: The covariance matrix mask (diagonal elements)
0,0,0,0,0........................0

Note that line #17 may not appear for some systems.

Health & Safety File

Data is written every time there is a subsystem status change. There will a indication of the new status and a reason why.

Time tag (DOY, Hour, Minute)
char NewStat
char subsystem and reason [80]

Circular buffer Debug File Format

This file will contain both the Angular Buffer and the Range Buffer parts of the Circular Buffer. The file is binary with format:

```
------------------------------------------------------------------------
"CIRCULAR BUFFER - RANGE"
struct RngCircBuf{
double RngT0SOY
struct RngElem[240000];
};
"CIRCULAR BUFFER - ANGLE"
struct AngCircBuf{
double AngT0SOY
struct AngElem[12000]
};
"END CIRCULAR BUFFER"
------------------------------------------------------------------------
```

where AngT0SOY is the starting time of the Angular Buffer and RngT0SOY is the starting time of the Range Buffer (ie all I2K values are referenced to this time). The following represent the data formats for the AngElem and the RngElem entries:

```
//  Angular Buffer: All angles are in degrees

struct AngElem{
long   I2K;          /* # of 2KHz interval past the buffer start time */
long   tbias;        /* time bias (usec) */
double azenc;     /* azimuth position - measured */
double elenc;     /* elevation position - measured */
double azbias;    /* total azimuth bias */
double elbias;    /* total elevation bias */
double azcmd;    /* commanded azimuth (predicted + biases) */
double elcmd;     /* commanded elevation (predicted + biases) */
double azrate;    /* azimuth rate (deg/sec) - measured */
double elrate;     /* elevation rate (deg/sec) - measured */
double azqd;      /* azimuth bias from Quadrant Detector */
double elqd;       /* elevation bias from Quadrant Detector */
double azscan;    /* azimuth scan bias */
double elscan;     /* elevation scan bias */
double azmanual;  /* manual bias input from RAT */
double elmanual;  /* manual bias input from RAT */
double azmm;      /* mound model calculation */
double elmm;      /* mound model calculation */
```

```
double elref;     /* refraction correction calculation */
double azptahead; /* transmit point ahead angle (difference from receive) */
double elptahead; /* transmit point ahead angle (difference from receive) */
double azdome;    /* dome measured location */
};
```

```
// Ranging Buffer: all integers are in picoseconds - all reals are in microseconds
// All data, except where specified, is associated with THIS interval's fire

struct RngElem{
long   I2K;       /* # of 2KHz interval past the buffer start time */
long   dtF;       /* delta fire time (for this interval's fire) in psec */
double Rp;        /* predicted range for this fire (usec) */
double Rpdot;     /* predicted range rate for this fire (usec) */
long   tblank;    /* blanking time for this interval (psec) */
long   MR;        /* number of intervals to this fire's expected return */
long   dtE;       /* expected time into THIS interval for return from previous fire (psec) */
long   MF;        /* # of interval back to fire for return(s) in this interval */
long   dtR[4];    /* measured range delta(s) from start of interval for returns from this fire (psec) */
double R[4];      /* measured roundtrip range for this fire (usec)*/
char   Q[4];      /* tag indicating quadrant for each return (1,2,3,4, or 0) */
char   N[4];      /* flag indicating signal(=1), noise(=2), or no data(=0) */
double rbias;     /* range bias (usec) */
long   tbias;     /* time bias (usec) */
long   rwin;      /* range window width (psec) */
double dRref;     /* refraction correction (usec) */
};
```

Message File

This file is similar to the New Controller’s message file. Data is written (one line per event) whenever a significant event happens, including any error conditions that occur, and any major decisions that were made by the POP Algorithms. The file is closed at the end of day. The form is:

date/time event description

Examples of significant events are:

Dome OPENED or CLOSED
Laser ENABLED or DISABLED.
Laser ON or OFF.
Simulation of MOUNT ON.
Barometer error flag ON.
Star camera image recorded to file.
System status changed to ORANGE.
Now tracking LAGEOS.
Sky too cloudy, STARLETTE pass rejected.
Starcal STARTED or COMPLETED.
Test MODE entered.
Security alarm #1
ICC not responding. etc...

This file can also contain debug information, if debug output is turned on. More detailed information will then be written to this file, but the format will remain the same. The normal operational mode, however, will be to write only significant events and decisions to this file.

13) Shared Memory

13.1) ICC <-> POP

```
;/* updated: 6/30/1998 jlfm */
/* updated: 7/22/1998 jwc -- changed all int's to short's, rearranged some
 *                      things for better word alignment
 * updated: 8/20/1998 jwc -- more mods
 */
/* -----------------------------------------------------------------------
 * ICC <--> POP Shared memory
 * These shared memory items reside in the Dual Port RAM on the VME bus side
 * of the Bit-3 Adaptor Card at address 0x400000, the size will be
 * determined later, but should never exceed 2MB as there is only 4MB
 * on the Dual Port Ram, and DAN will need a chunk of memory also.
 *
 * It is imperative that word alignment be maintained for the ICCPOP/POPICC
 * transfers, 'dummy' variables  are used to maintain the alignment, these
 * may be adjusted accordingly.
 */
#ifndef __ICCPOPSHM_H
#define __ICCPOPSHM_H
struct POPBuffer {
   short POPTimeInterval; /* 0 to 1999 */
   short BlankTime;
   long RngWin;
   long RngDelay;
   char RngMode[2];    /* bit 0 -- 0==> no ranging (overides all bits)
                        * bit 6 -- 1==> randomize (only if bit 7 is on)
                        * bit 7 -- 1==> simulate data
                        */
   char LaserMode[2];  /* bits 0 & 1  00=Laser disable
                        *             01=Laser enable
                        *             11=Laser ON
                        *             10=will be ingnored (treated as disable)
                        */
   long PRFvoltcmd;
   long PRFcmd;       /* in thousandths of Hertz 1990-2010 */
   short POPMntInterval;
   char MntMode[2];
                /* bit 0 -- 0==> no mount output or input (overides all bits)
                 *       -- 1==> normal operation (use other bits)
                 * bit 1 -- 1==> output to mount next interval
                 * bit 6 -- 1==> randomize (only if bit 7 is on)
                 * bit 7 -- 1==> simulate data
                 */
   long MntAzcmd;
   long MntElcmd;
   long MntAzvel;
   long MntElvel;
```

```
   long MntAzacc;
   long MntElacc;
   short CamExpTime;  /* 0 to 10000 milliseconds */

   char CamMode; /* bit 0 -- 1==> take camera data */

   char ICCMode; /* bit 0 -- 0==> Real-Time (Operational Mode)
                  * bit 0 -- 1==> Non Real-Time (Diagnostic Mode)
                  */
   char DiagFlag[2];
   char dummy[2]; /* this is to keep the buffer a multiple of 4 for easier
                   * transfer to POP, if any additions or subtractions are
                   * made this may be adjusted accordingly
                   */
};

struct smpopicc {
/* Buffer availability indicator at Dual Port RAM offset 200.
 */
/*  POPiDataAvailFlag (i=0,1) are the variables used to determine
 *  whether POP buffer #i has data available to be read by ICC.
 *  POP keeps its write buffer indicator (0 or 1) in a variable
 *  outside of SH.MEM.
 */
/* Following line changed: 6-30-1998 */
   short POP0DataAvailFlag;     /* 1==> buffer #0 is ready to read. */
/* Following line changed: 6-30-1998 */
   short POP1DataAvailFlag;     /* 1==> buffer #1 is ready to read. */
   struct POPBuffer POPbuf[2];
};

struct ICCBuffer {
   short ICCTimeInterval; /* 0 to 1999 */
   short CtrTimingError;  /* 0 ==> No Error. */

   char TimingStat;     /* bit 0 -- 1==> real-time.
                         * bit 6 -- 1==> error
                         */
   char RngStat;        /* bit 0 -- 1==> ranging is occurring (real or sim).
                         * bit 1 -- 0==> Picosecond Time Analyzer
                         *           1==> Event Timer.
                         * bit 6 -- 1==> error.
                         * bit 7 -- 1==> simulated data.
                         */
   char MntStat;        /* bit 0 -- 0==> mount drive is on (real or sim).
                         * bit 6 -- 1==> error.
                         * bit 7 -- 1==> simulated data.
                         */
   char LaserStat;      /* bit 0 -- 1==> Laser is ON.
                         * bit 6 -- 1==> error.
                         */
```

```
   char CamStat;        /* bit 0 -- 1==> Camera data is currently being taken.
                        * bit 6 -- 1==> error.
                        * bit 7 -- 1==> simulated data.
                        */

   char XferStat;       /* bit 5 -- 1==> BIT-3 error
                        * bit 6 -- 1==> POP transfer error.
                        */
   char psecAnalStat;
   char CamacStat;
   char unidexStat[2];
/*   char dummy; */          /* This is here to keep word alignment */
   short MntInterval;
   short RngInterval;
   char TAGbits[4];
   long psecAnal[4];
   short Rctr10Mhz;
   short Fctr10Mhz;
   long Rdelt10Mhz;
   long Fdelt10Mhz;
   long Rdelt2Khz[4];   /* fixed point TBD */
   long Fdelt2Khz;      /* fixed point TBD */
   long PRFvolt;
   long azpos;
   long elpos;
   char MntLimitsw;     /* Mount limit switch flags
                        * bit 0 = 1 ==> azimuth home position
                        * bit 1 = 1 ==> elevation home position
                        * bits 2-7 TBD
                        */
   char cablewrap;      /* cable wrap flags
                        * bit 0 = 1 ==> azimuth cable wrap limit
                        * bit 1 = 1 ==> elevation cable wrap limit
                        * bits 2-7 TBD
                        */

};

struct smiccpop {
/*  ICCiDataAvailFlag (i=0,1) are the variables used to determine
 *  whether ICC buffer #i has data available to be read by POP.
 *  ICC keeps its write buffer indicator (0 or 1) in a variable
 *  outside of SH.MEM.
 */
/* Following line changed: 6-30-1998 */
   short ICC0DataAvailFlag;  /* 1==> buffer #0 is ready to be read. */
/* Following line changed: 6-30-1998 */
   short ICC1DataAvailFlag;  /* 1==> buffer #1 is ready to be read */
   short CamAvailFlag;  /* 1==> camera buffer ready for pickup */
   short DiagAvailFlag; /* 1==> diagnostic data is ready for pickup */
   struct ICCBuffer ICCbuf[2];
```

```
};

/* Camera Data
 */
struct smicccam {
   char CamData[58564];
};

/* Diagnostic Data
 */
struct smdiag {
   char dummy[2];
};


#define SHMIPSIZE (sizeof(struct smiccpop))
#define SHMPISIZE (sizeof(struct smpopicc))
#define SHMICSIZE (sizeof(struct smicccam))
#define SHMDGSIZE (sizeof(struct smdiag))
#endif
```

13.2) POP <-> DAN

```
/*
;*******************************************************************************
;  File:                    popdanshm.h
;
;  Project:    SLR 2000
;
;  Author:     Randall L. Ricklefs
;              Univ. of Texas / McDonald Observatory
;  Date:       February, 1998
;
;  Function:   Provide shared memory description for interface between
;              Data Analysis Computer (DAN) and Pseudo-Operator Computer (POP)
;              (Also used in DAN's Remote Access Terminal (RAT) Ratsnest server
;              program.)
;
;  History:
;    Revision 1.6  1998/10/09 11:49:00 jwc
;    added bitwise variables & H&S status from DAN
;
;    Revision 1.5  1998/08/20 15:20:21 jwc
;    other mods
;
;    Revision 1.4  1998/07/22 13:23:55 jwc
;    changed int's to shorts
;
;    $Log:            shared_mem_DANPOP.h,v $
;    Revision 1.3  98/06/17  14:08:54  rlr
;    Cleaned up comments.
;
;    Revision 1.2  1998/04/08 12:22:25  rlr
;    First usable version
;
;    Revision 1.1  1998/02/27 13:23:04  rlr
;    Initial revision
;
;
;*******************************************************************************
*/
#ifndef __POPDANSHM_H
#define __POPDANSHM_H
static char rcDANRATsid[] = "$Id: popdanshm.h,v 1.2 1998/04/08 12:22:25 rlr Exp rlr $";


struct smpopdan {

/* Time */
   short year;
   short doy;
   long sod;
```

```
   short int2khz;
   /* char TimeOvrdFlag; */
    /* Bits are:
     *   0=simulation time from CPU
     *   1=simulation time from override (mutually exclusive with above)
     *   2=use clock time offset
     *   3=ignore GPS-computed offset
     */
     unsigned SimTimeCPU: 1;
     unsigned SimTimeOVRD: 1;
     unsigned ClkTimeOffset: 1;
     unsigned IgnGPSOffset: 1;
     unsigned :4;   /* spares */

/* Actual mount and dome position */
   double MntAzRaw;
   double MntElRaw;
   double DomeAzRaw;
   short DomeShutter;        /* 0=closed; 1=open */
   double sc_az[2];      /* sky camera azimuth corners */
   double sc_el[2];      /* sky camera elevation corners */

/* various flags */
   char SysMode[3];      /* Tracking mode - 1 byte; Object - 2 bytes */

   /* char FlagByte1; */ /* updated at 1 Hz Transfer Rate */
       /* bits 0&1: operation 00, testing 01, diagnostics 10
        * note: operation is when RAT is not connected.
        *      Testing is when RAT is connected, running operational sequence
        *      Diagnostics is when not running operational sequence
        *      (RAT not connected)
        * bit 2: Order DAN shutdown
         * bit 3: POP active
         * bits 4-7: spare
         */
     unsigned PopSystemMode: 2;
     unsigned DanShutdown: 1;
     unsigned PopActive: 1;
     unsigned :4;  /* spare */
   /* char FlagByte2; */ /* updated at 1 Hz Transfer Rate */
        /* bit 0: type of ranging system 0==> Picosecond Analyzer
         *                              1==> Event timer
         * bit 1: request sky clarity (This bit will go on and be cleared by
         *                          POP and only POP, DAN will set the
         *                          corresponding bit in the DanFlagByte2
         *                          to signal that the data is available.)
        * bits 2-7: spare
         */
     unsigned TypeRanging: 1;
     unsigned RequestSkyClarity: 1;
     unsigned: 6; /* spares */
```

```
  /* char FileFlagByte1; */ /* updated at 1 Hz Transfer Rate */
     /* bit 0: Data file ready
      * bit 1: Star Camera file ready
      * bit 2: spare
      * bit 3: spare
      * bit 4: Starcal file ready
      * bit 5: spare
      * bit 6: Decision file ready
      * bit 7: spare
      */
    unsigned DataFileReady: 1;
    unsigned StarCameraFileReady: 1;
    unsigned: 2; /* spares */
    unsigned StarcalFileReady: 1;
    unsigned: 1; /* spare */
    unsigned DecisionFileReady: 1;
    unsigned: 1;
  /* char FileFlagByte2; */ /* updated at 1 Hz Transfer Rate */
     /* bits 0-7: spare
      */
    unsigned spare:1;
    unsigned: 7; /* spares */
  /* char SysExcept[2]; */
     /* bit 0: In sun avoidance
      * bit 1: In mask avoidance
      */
    unsigned SunAvoidance: 1;
    unsigned MaskAvoidance : 1;
    unsigned: 14; /* spares */
  /* char SubSysPwr[2]; */
    unsigned: 16; /* spares */

/* ICC system status */
  char ICCSysStat;                          /* color */
  char ICCReasonInstr[2];
  char ICCReasonText[16][80];

/* POP system status */
  char POPSysStat;                          /* color */
     /* +1 = green => OK
      *  0 = white => not up
      *  <0 => not OK
      * -1 = yellow
      * -2 = orange
      * -3 = red
      * -4 = black
      */
  char POPReasonInstr[8];
  char POPReasonText[64][80];

/* Subsystem status (color) */
```

```
   char DomeStat;
   char LaserStat;
   char MountStat;
   char RangingElecStat;
   char StarCamStat;

/* Current file names */
   char DataFileName[80];
   char StarCamFileName[80];
   char DebugFileName[80];
   char StarCalFileName[80];
   char MessageFileName[80];
   char DecisionFileName[80];
};

struct smdanpop {
/* System Status from Health and Safety */
   char HandSStat;
       /* +1 = green => OK
        *  0 = white => not up
         * <0 => not OK
        * -1 = yellow
        * -2 = orange
        * -3 = red
        * -4 = black
        */
/* DAN system status */
   char DANSysStat;                    /* color */
   char DANReasonInstr[4];
   char DANReasonText[32][80];

/* various flags */
   /* char DANFlagByte1;  updated at 1 Hz Transfer Rate
        * bits 0&1: operation 00, testing 01, diagnostics 10, as above
        * bit 2: DAN active
        * bit 3: New Predictions
        * bit 4: New Schedule
        * bit 5: Prepare Shutdown
        * bit 6: Shutdown
        * bit 7: RAT attached
        */
     unsigned DanSystemMode: 2;
     unsigned DanActive: 1;
     unsigned UpdPredictions: 1;
     unsigned UpdSchedule: 1;
     unsigned PrepareShutdown: 1;
     unsigned Shutdown: 1;
     unsigned RatAttached: 1;
   /*char DANFlagByte2;   updated at 1 Hz Transfer Rate
        * bits 0: spare
         * bit 1: Sky clarity data ready (See Flagbyte2 in popdan
```

```
        * bits 2-7: spare
       */
    unsigned: 1; /* spare */
    unsigned SkyClarityDataReady: 1;
    unsigned: 6; /* spares */
  char DANSimFlag[8];
      /* Things to simulate (from RatsNest)
       * (1) simulate ranging system and laser
       * (2) simulate mount drive
       * (3) simulate star camera
       */
  char DANTestFlag[8];
      /* Things to override
       */
  char DANDiagFlag[4];
      /* Diagnostic tests to run
       */
  double temp;        /* temperature  (celcius) */
  double barp;        /* barometric pressure (millibars) */
  short hum;          /* humidity (percentage) */
  short wind_spd;     /* wind speed (meters/second) */
  short wind_azm;     /* wind direction (0-359) */
  short visibility;   /* horizontal visibility (kilometers) */
  char precip[4];     /* precipation type, See precipcodes.txt for definition*/
  char SkyClarity;    /* 0 - no data, 1 - clear, 2 - hazy, 3 - cloudy */
};

#define SHMPDSIZE (sizeof(struct smpopdan))
#define SHMDPSIZE (sizeof(struct smdanpop))

#endif
```

13.3) RAT

```
/*
;*******************************************************************************
;  File:          ratshm.h
;
;  Project:    SLR 2000
;
;  Author:     Randall L. Ricklefs
;              Univ. of Texas / McDonald Observatory
;  Date:       February, 1998
;
;  Function:   Provide shared memory description for interface between
;              Data Analysis Computer's (DAN) Remote Access Terminal (RAT)
;              server "Ratsnest" and the RAT graphical user interface "Ratgui"
;              on the RAT laptop computer.
;
;  History:
;    $Log:        ratshm.h,v $
;    Revision 1.3  98/06/17  14:08:03  rlr
;    Cleaned up comments.
;
;    Revision 1.2  1998/04/08 12:23:01  rlr
;    First usable version - Thanks to Jan, Jack, Brion.
;
;    Revision 1.1  1998/02/27 13:23:04  rlr
;    Initial revision
;
;
;*******************************************************************************
*/
#ifndef __RATSHM_H
#define __RATSHM_H
static char rcRATsid[] = "$Id: ratshm.h,v 1.2 1998/04/08 12:23:01 rlr Exp rlr $";

struct smrat
{
/* Time */
   char TimeOvrdFlag;
    /* Bits are:
     *   0=simulation time from CPU
     *   1=simulation time from override (mutually exclusive with above)
     *   2=use clock time offset
     *   3=ignore GPS-computed offset
     */

   int year_ovrd;
   int doy_ovrd;
   double sod_ovrd;            /*in seconds*/
   double clock_time_offset;          /*in seconds*/
```

```
   double pred_time_offset;          /*in seconds*/

/* Filenames */
   char SkyCamFileName[80];
   char SecurityCamFileName[80];
   char HandSFileName[80];

/* Picture Ready Flags */
   int SkyCamFileReady;
   int SecurityCamFileReady;
   int StarCamFileReady;

/* Subsystem Status: color */
   char SecurityCamStat;
   char SecuritySysStat;
   char SkyCamStat;
   char InetCommStat;
   char PhoneCommStat;
   char SecurityFlags[2];

/* Schedule */
   char SchedControlFlag;
    /* Bits are:
     *   0  1==>Override automatic schedule
     */
   long SchedWhatToDo;
   int SchedStartDoy;
   int SchedStartTime;
   int SchedDuration;
   long SchedObject;

/* Star */
   int StarOvrdMode;   /* -1 to automatically choose; 0 nothing; > 0 fk5 # */
   char StarControlFlag1;
    /* Bits are:
     *   0=StarHold (1==>HOLD)
     *   2=StarFocusCorr (1==>No focus correction)
     *   3=StarUseImageBiases (1==>Do NOT use camera biases)
     *   4=Auto-increment off (when = 1)
     *   5=Auto-starcal off (when = 1)
     *   6=StarManualRecordBiases (1==>record; POP must turn this bit off)
     *   7=StarGenSoln (1==>generate solution; POP must turn this bit off)
     */
   char StarControlFlag2;

/* Dome */
   char DomeControlFlag;
    /* Bits are:
     *   0=DomeFixShutter (1==>fix shutter)
     *   1=DomeAzBias (1==>bias Dome AZ using value below)
     *   2=DomeAzFix (1==>fix dome position at value given below)
```

```
    *   7=Shutter OPEN/CLOSED (1==>OPEN)
    */
   int DomeAzBiasValue;
   int DomeFixAzValue;

/* Search */
   int SrchAngFlag;
    /* Bits are:
    *   0=SrchAngLengthOvrdFlag
    *   1=SrchAngStepOvrdFlag
    *   2=SrchAngStepDwellFlag
    *   4=SrchRngLengthOvrdFlag
    *   5=SrchRngStepOvrdFlag
    *   6=SrchRngStepDwellFlag
    */
   long   SrchAngLengthOvrdValue;
   double SrchAngStepOvrdValue;
   long   SrchAngStepDwellValue;   /* # of frames */
   long   SrchRngLengthOvrdValue;
   double SrchRngStepOvrdValue;
   long   SrchRngStepDwellValue;   /* # of frames */


/* Decisions */
   char DecnOvrdFlag1;
    /* Bits are (1==>IGNORE):
    *   0=DecnMaskOvrd
    *   1=DecnSunAvoidIgnore
    *   2=DecnHSStatusOvrd
    *   3=DecnMountStatErrIgnore
    *   4=DecnLaserStatErrIgnore
    *   5=DecnDomeStatIgnore
    *   6=DecnFocusNotCorrectIgnore
    */
   char DecnOvrdFlag2;
    /* Bits are:
    *   0=DecnStarCamStatErrIgnore
    *   1=DecnRngElecStatErrIgnore
    */

   int HSStatusOvrdValue;
    /*      Should we call this instead: SysStatusOvrd? */

   char DecnControlFlag;
    /* Bits are:
    *   0=DecnAngScanFlag (1==>do NOT perform an angular scan)
    *   1=DecnUseQDBiasFlag (1==>do NOT use QD angular biases)
    *   2=DecnRngScanFlag (1==>do not perform a range scan)
    *   3=DecnLaserFlag (1==>control laser from next bit)
    *   4=Laser ON/OFF (1==>ON)
    *   5=Fit time bias (1==>OFF)
```

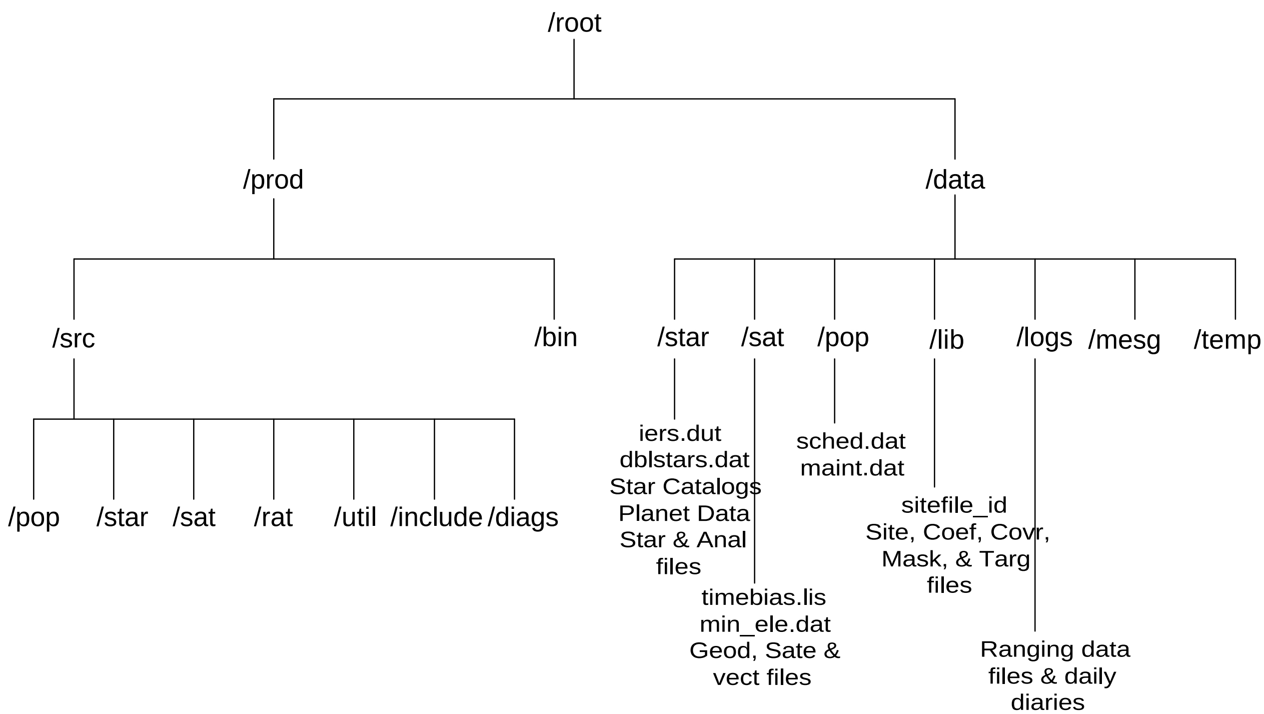
```
    */

/* Mount */
   char MntControlFlag;
    /* Bits are:
         0=MntAzBiasFlag (1==>add bias given below to AZ)
         1=MntElBiasFlag (1==>add bias given below to EL)
         2=MntFixAzFlag (1==>move AZ to fixed value given below)
         3=MntFixElFlag (1==>move EL to fixed value given below)
         4=MntFixAzRateFlag (1==>drive AZ at fixed rate: use values below)
         5=MntFixElRateFlag (1==>drive EL at fixed rate: use values below)
     */
   double MntAzBiasVal;
   double MntElBiasVal;
   double MntFixAzVal;
   double MntFixElVal;
   double MntFixAzRateVal;
   double MntFixElRateVal;
   double MntStartAzVal;
   double MntStopAzVal;
   double MntStartElVal;
   double MntStopElVal;

/* Ranging */
   char RngControlFlag;
    /* Bits are:
     *   0=RngBiasFlag (1==>bias range with value below)
     *   1=RngFixDelayFlag (1==>fix range with value below)
     *   2=RngFixWinFlag (1==>fix range window with value below)
     */
   double RngBiasVal;
   double RngFixDelayVal;
   double RngFixWinVal;

/* Sensors */
   char SensorOvrdFlag[8];
    /* Bits are (1==>override using values below):
     *   Byte#1
     *   0=SensTempOvrdFlag
     *   1=SensBaroOvrdFlag
     *   2=SensHumidOvrdFlag
     *   3=SensPercipOvrdFlag
     *   4=SensVisOvrdFlag
     *   5=SensWindSpdOvrdFlag
     *   6=SensWindDirOvrdFlag
     *   7=SensSkyClarOvrdFlag
     */
   double SensTempOvrdVal;
   double SensBaroOvrdVal;
   double SensHumidOvrdVal;
   double SensPercipOvrdVal;
```

```
  double SensVisOvrdVal;
  double SensWindSpdOvrdVal;
  double SensWindDirOvrdVal;
  int    SensSkyClarOvrdVal;
/** To be Defined
  int SensSecAlarmOvrdFlag[xx];
  int SensSecAlarmOvrdVal[xx];
  int SensSysTempOvrdFlag[xx];
  double SensSysTempOvrdVal[xx];
  int SensSysVoltOvrdFlag[xx];
  double SensSysVoltOvrdVal[xx];
**/

/* Sensors */
  int CamSkyFlag;                /* 1=> request images */
  int CamStarFlag;               /* 1=> request images */
  int CamSecFlag;                /* 1=> request images */
  long CamSkyInterval;       /* interval for requested images */
  long CamStarInterval;      /* interval for requested images */
  long CamSecInterval;       /* interval for requested images */
  char CameraFlag;
   /* Bits are:
    *   0=change star camera exposure time
    */
  double CamStarExpTimeVal;
  double SimAmbTmp;  /* simulated ambient temperature for IR camera */

/* Miscellaneous (Other) */
  char MiscControlFlag1;
   /* Bits are:
    *   0=OthAltMountMod (1==>use alternate mount model:this needs to be worked out!)
    *   1=OthFocusBiasFlag (1==>bias focus setting with value below)
    *   2=OthFixFocusFlag (1==>fix focus setting at value given below)
    *   3=OthPointAheadBiasFlag (1=>bias point-ahead with value below)
    *   4=OthFixPointAheadFlag (1==>fix point-ahead with value given below)
    *   5=OthWatchdogFlag (1==>turn OFF Watchdog timer)
    *   6=OthPRFOvrdFlag (1==>fix PRF at value given below)
    *   7=OthDbgCB (1==> circular buffer debugging)
    */
  long OthFocusBiasVal;
  long OthFixFocusVal;
  long OthPtAheaddBiasVal;
  long OthFixPtAheadVal;
  long OthFixBlankTimeVal;
  long OthPRFOvrdVal;

  char MiscControlFlag2;

  /* Some other flags... */
  char WaitingForSuperFrame;
  char SuspendBiasCalcs;
```

```
};

#define SHMRSIZE (sizeof(struct smrat))
int     smrid;                  /* smrat */
char    *smrptr;
#endif
```

14)Directory Structures

14.1) POP



The standard file naming conventions for the Star, Anal, Coef, Covr, Mask, Targ, and Vect files is as follows:

XXXX_sNNyNNNNdNNNtNNNN.ext

where XXXX denotes the file type and the N's denote the station ID the year, the day of year and the time. An extension will be applied if applicable, i.e for Star, either .gbl or .kal.

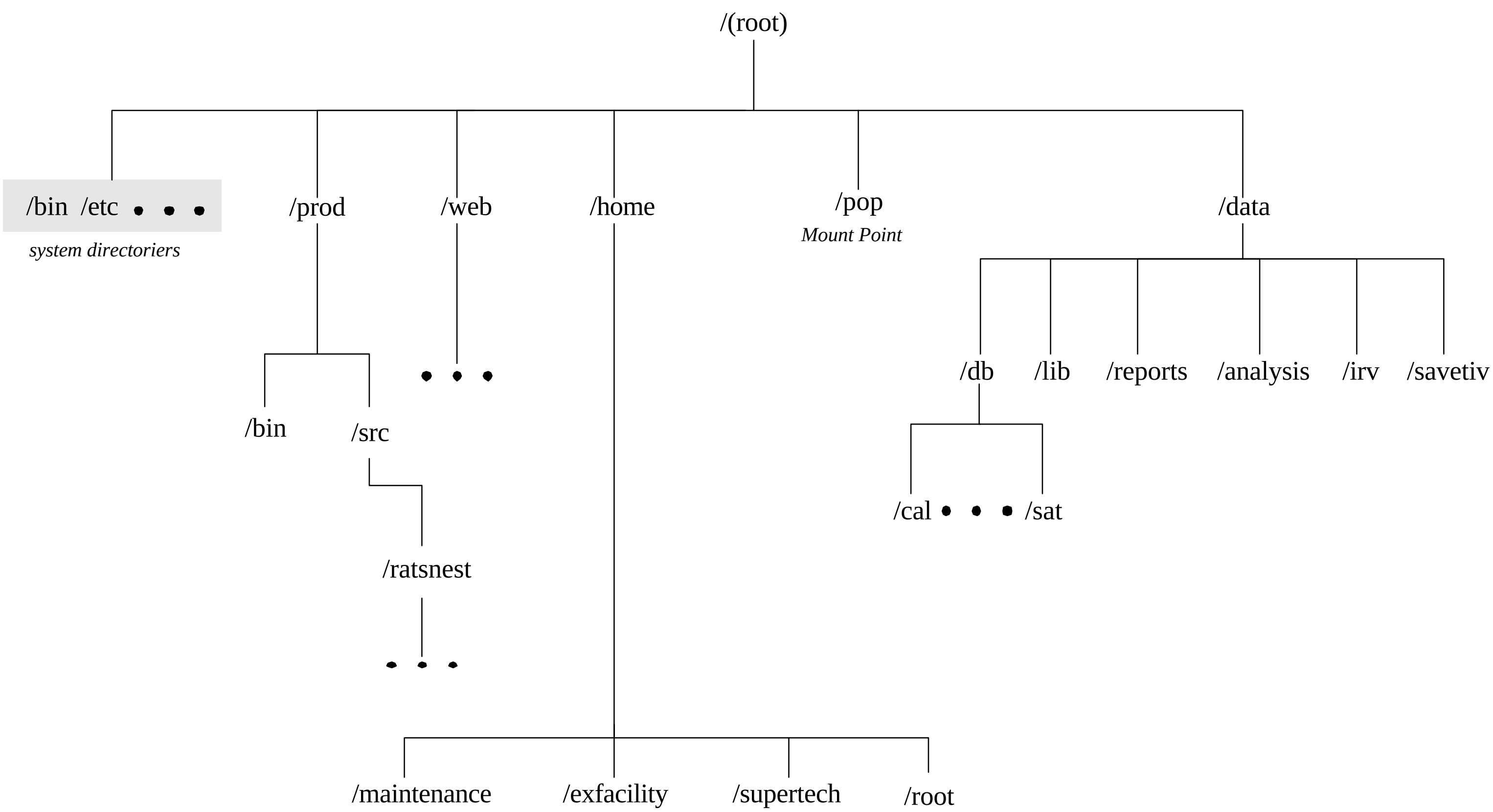
The message and debug files will have a similar naming convention as noted in the box to the left, but use 'mesg' and 'debg' respectively. The geodetics and satellite fileswill also use the same convention 'geod' and 'sate'. The camera files will use 'strc'.

The star catalogs and planetary data will have the following names:

FK5 catalog	-- FK5.dat
FK5 extension	-- FK5EXTN.dat
Messier catalog	-- MESSIER.dat
Planetary data	-- JPLEPH.dat

14.2) DAN

DAN'S DISK STRUCTURE



15) Remote Access Terminal(RAT)

15.1) Introduction

The RAT (Remote Access Terminal) is able to display basically all SLR2000-produced data files to various levels of complexity. All files can be displayed as text. Debug files allow spreadsheet-type display with some simple numerical manipulation. Log files, health and safety files, star calibration analysis files, post-fit residual files, and certain modes of debug files can be displayed with scatter or polar plots, as well. There is also real-time display of weather information, telescope position, and system status alarms.

In addition, the RAT allows overriding most important system parameters and decisions by interacting with its graphical user interface.

Communications for real-time displays (to the RAT) and commands (from the RAT) are accomplished with TCP sockets connected to the Ratsnest program on one of the host computers (POP for development, DAN for operations). Files are accessed via NFS mounts of the POP and DAN file systems on DAN.

15.2) Software components (Server)

Ratsnest provides access to real-time information

- Resides on Data Analysis Computer (DAN)
- Runs as a background information and command server whenever DAN is operating
- Passes DAN and Pseudo-Operator (POP) shared memory information to Ratgui via TCP sockets
- Acts on commands sent by Ratgui

15.3)Software Components (Client)

Ratgui provides graphical user interface to SLR2000

- Resides on laptop
- Can be run from anywhere across network (with security limitations)
- Receives real time status information from Ratsnest
- Accesses and displays data files after the fact via Network File System (NFS)

15.4) RAT's Graphical User Interface

15.4.1) Main window

See figure 15.4.1 (RatGui Main Window)

Menu Bar

- 'File' pull-down allows access to all file types
- 'Control' pull-down allows override of autonomous decisions

Control buttons

- Allows immediate execution of urgent commands

Alarm area

- Calls attention to abnormal situations

Command response scroll area

- Shows status of commands and overrides from ratgui

Status display

- Current time and date
- Current mount and dome positions
- Current weather information
- Hardware subsystem status flags

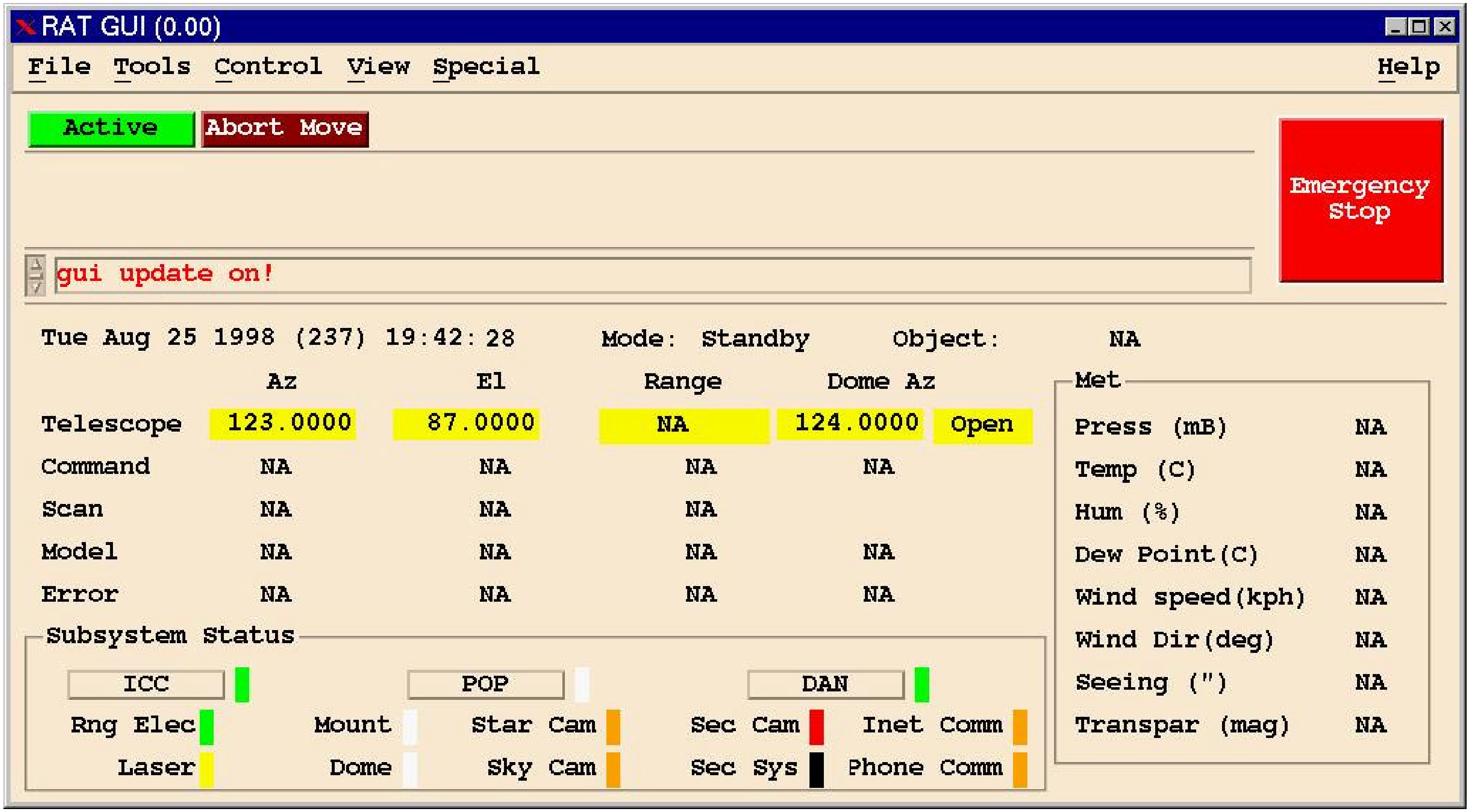


Figure 15.4.1 RatGui Main window

*5.4.2) File viewer*s

Debug file (see figure 15.4.2 a and 15.4.2 b)

- Contains range, mount, and dome sections
- Displayed as spreadsheet, allowing simple numeric manipulation
- Displayed as plots to show mount and dome performance

Decisions and Housekeeping files

- Text display of contents

FITS Image files (Star, Sky, and Security Cameras)

- Display and manipulate via XV, a common Unix image program

Health and Safety (see figure 15.4.2c)

- Each of the many data components can be plotted versus time
- Plotted quantities include such things as pressure, temperature, humidity, wind speed and direction and various internal voltages and temperatures

Log (raw data) (see figure 15.4.2 d)

- Plots of satellite ID, range gate, scan and O-C range residuals
- Plots of azimuth and elevation mount model and scan offsets, and angular residuals

Post-fit Range Residuals (see figure 15.4.2 e)

- Plot of O-C residuals of noise, data, and normal points

Star Calibration Analysis (see figure15.4.2 f)

- Total of 12 plots of error vs. axis, error vs. error, and polar error and model plots
- Can plot one or 2 data sets on one plot

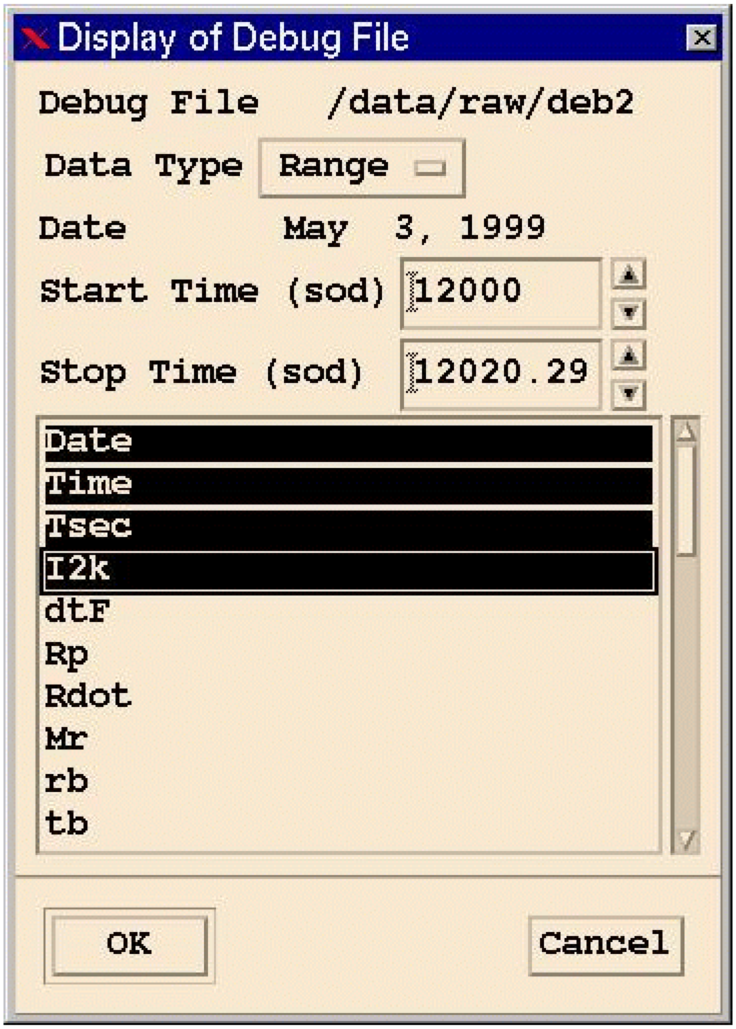


Figure 15.4.2 a- Selecting Debug File components

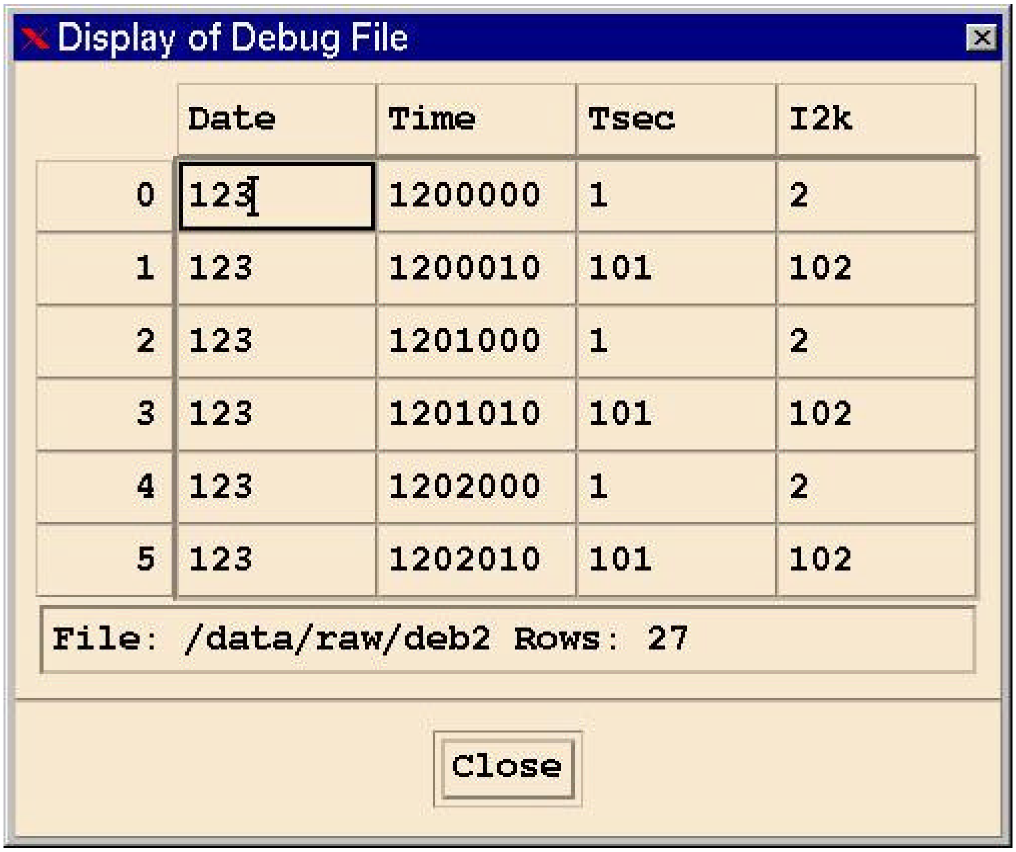


Figure 15.4.2 b- Spreadsheet Display of Debug file

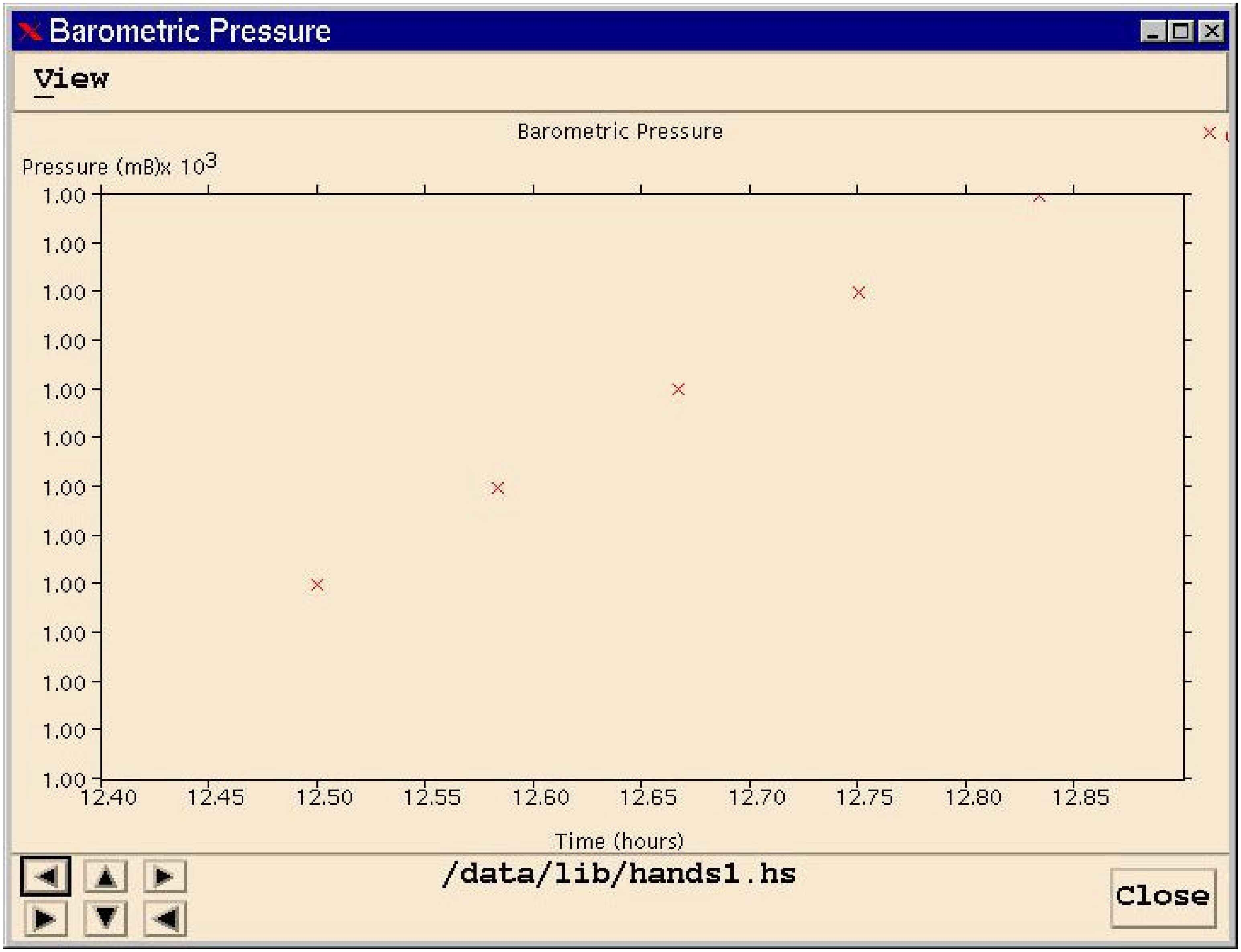


Figure 15.4.2 c- Health and safety display

Figure 15.4.2 d -Range Data Display

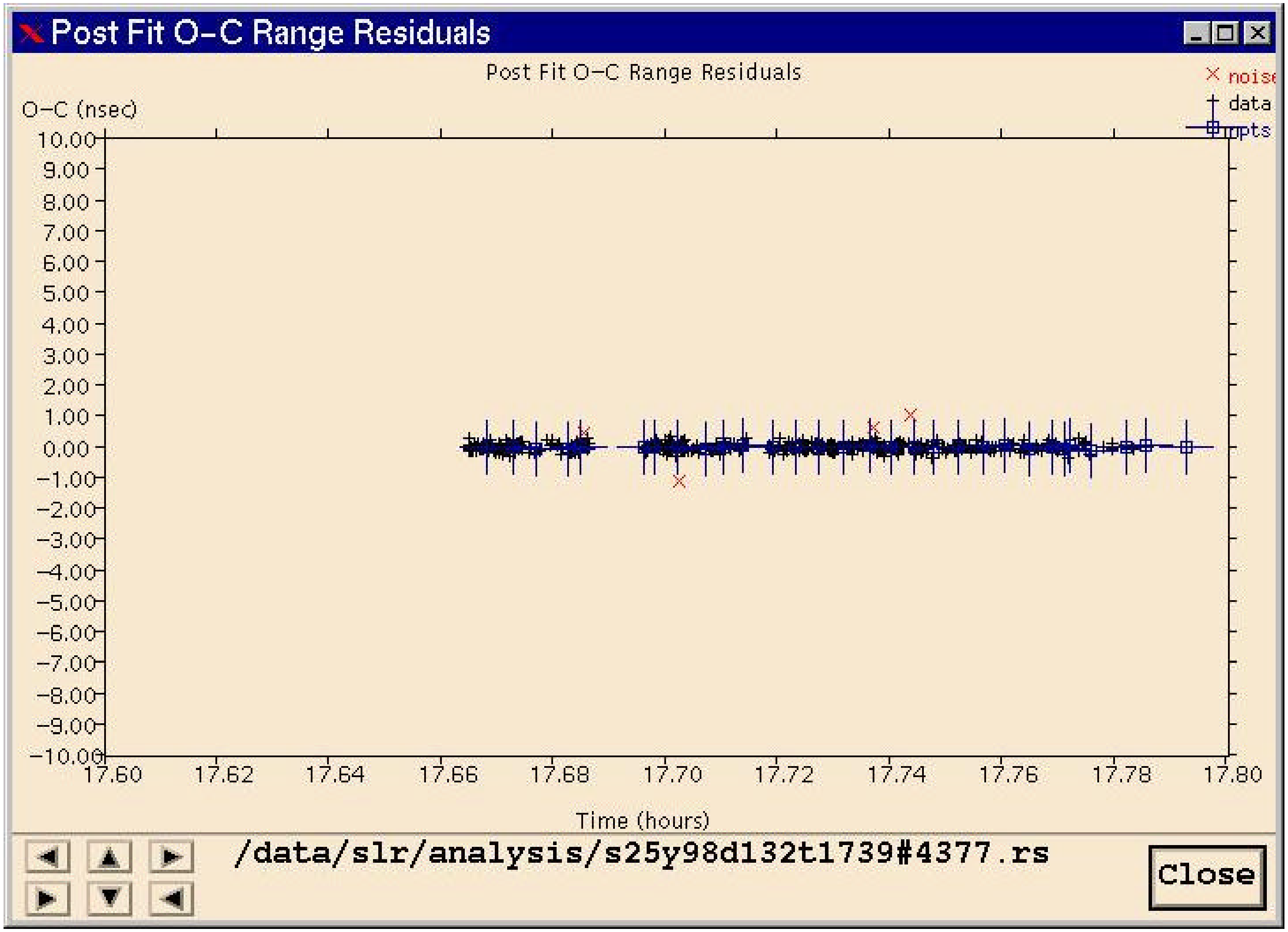


Figure 15.4.2 e - Postfit Range Residuals

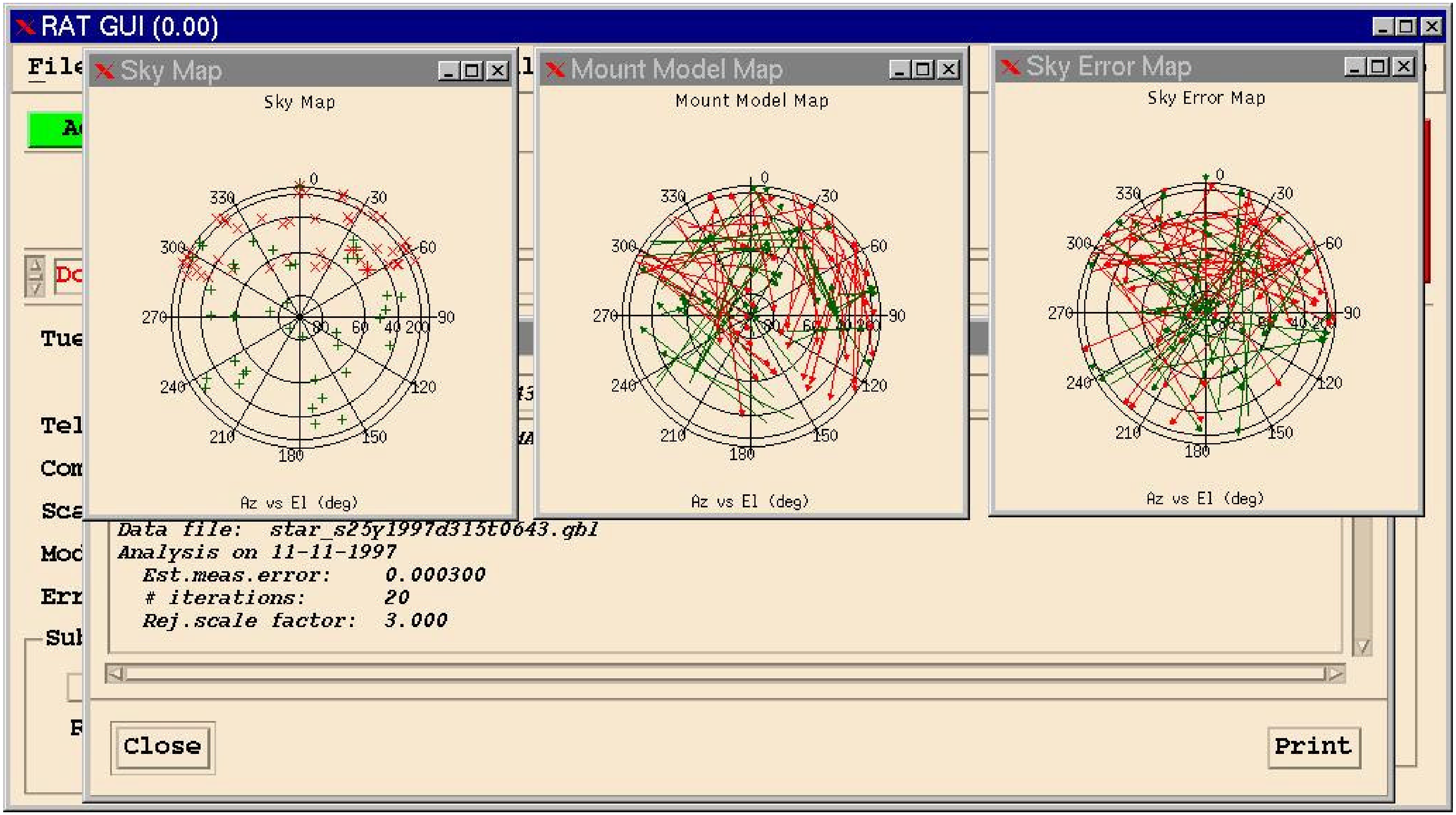


Figure 15.4.2 f- StarCal Analysis Results : 3 of 12 Possible Plots

15.4.3) controls and overrides

Overrides of the current values for various options and control flags toggling use of these overrides is provided through the Ratgui Control menu. This includes control and override of:

- time (see figure 15.4.3a)
  Set time of day, simulation time, time offsets and the like.
- schedule (see figure 15.4.3b)
  Select new target for a time slot.
- starcal (see figure 15.4.3c)
  Controls acquisition of mount modeling observations.
- dome
  Controls dome motion.
- search (see figure 15.4.3d)
  Overrides values of target search algorithm.
- decisions (see figure 15.4.3.e)
  Overrides system decisions.
- mount (see figure 15.4.3 f)
  Controls mount position and rate; allows for diagnostic tests.
- range
  Overrides range bias, delay, and window width.
- sensors
  Overrides met sensor values.
- cameras (see figure 15.4.3 g)
  Controls camera exposure times and the like.
- misc
  Overrides various functions not fitting into the other categories.

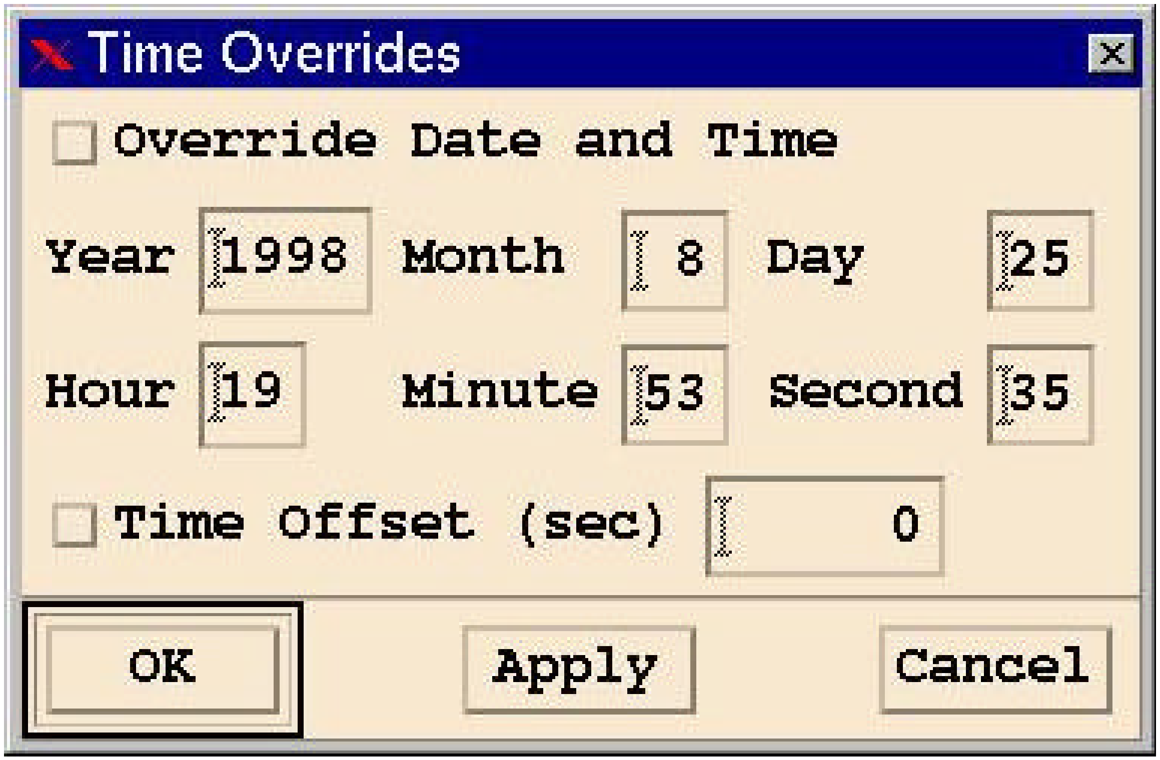


Figure 15.4.3a Ratgui time control and override

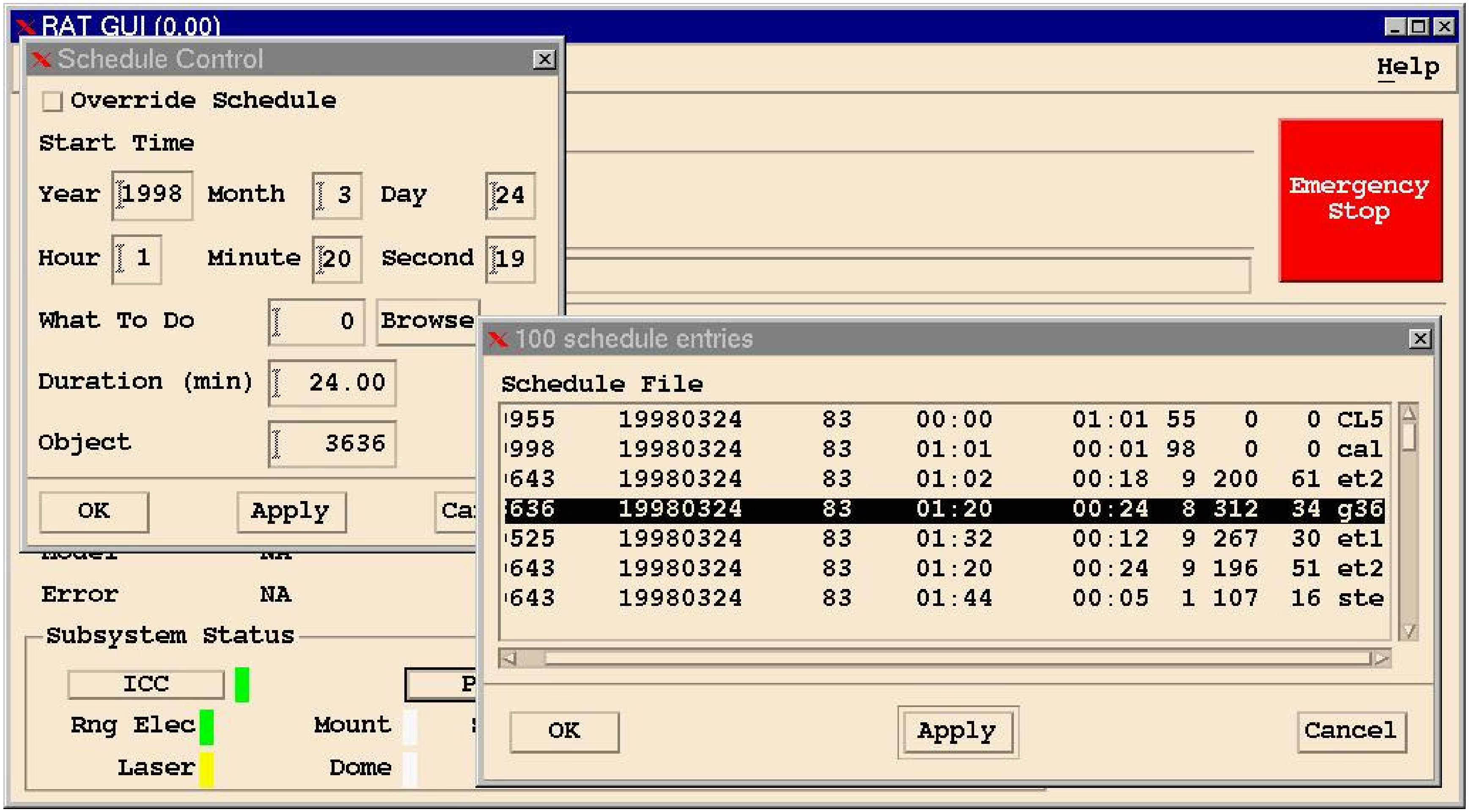


Figure 15.4.3b-Ratgui schedule control and override

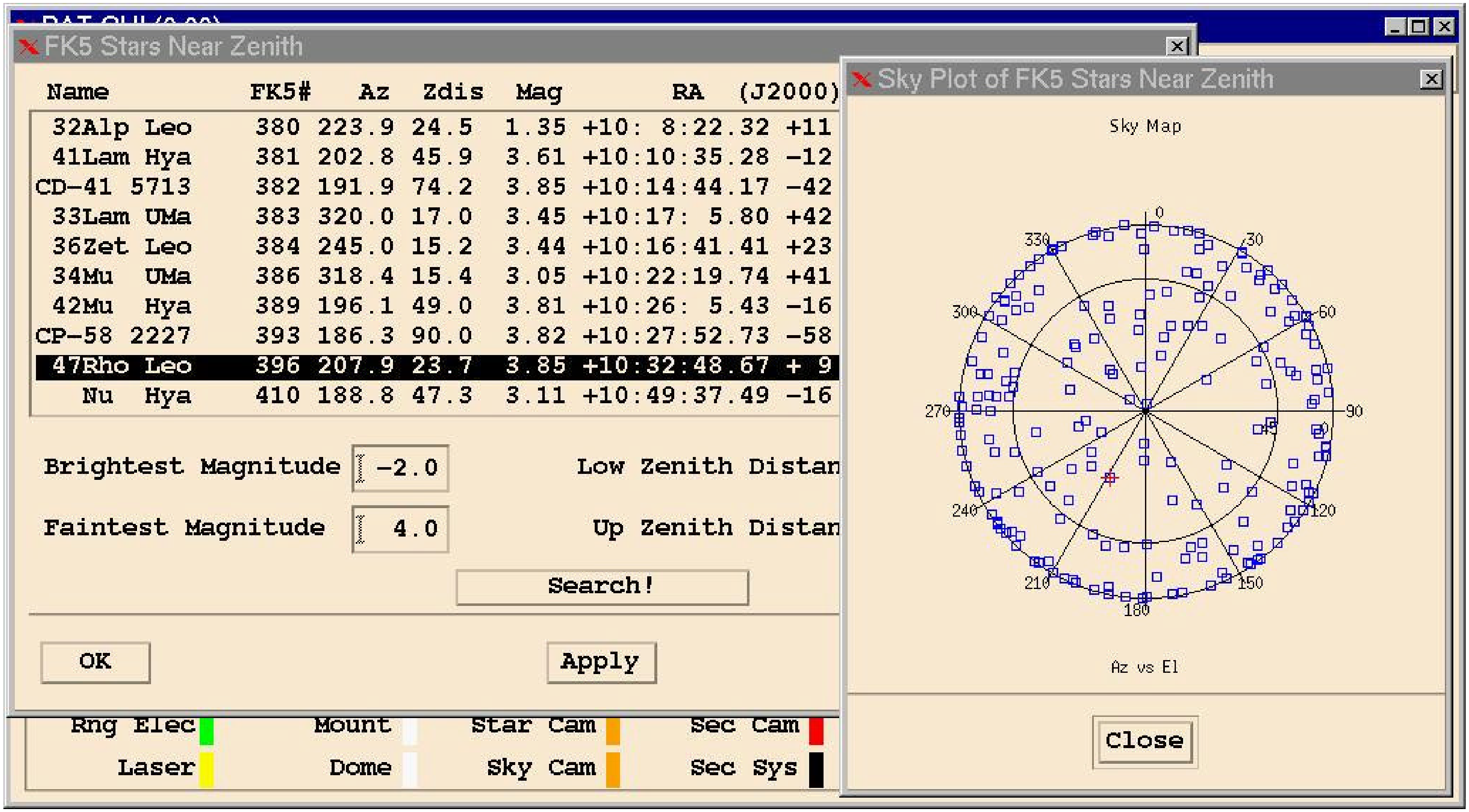


Figure 15.4.3c-Ratgui starcal control and override

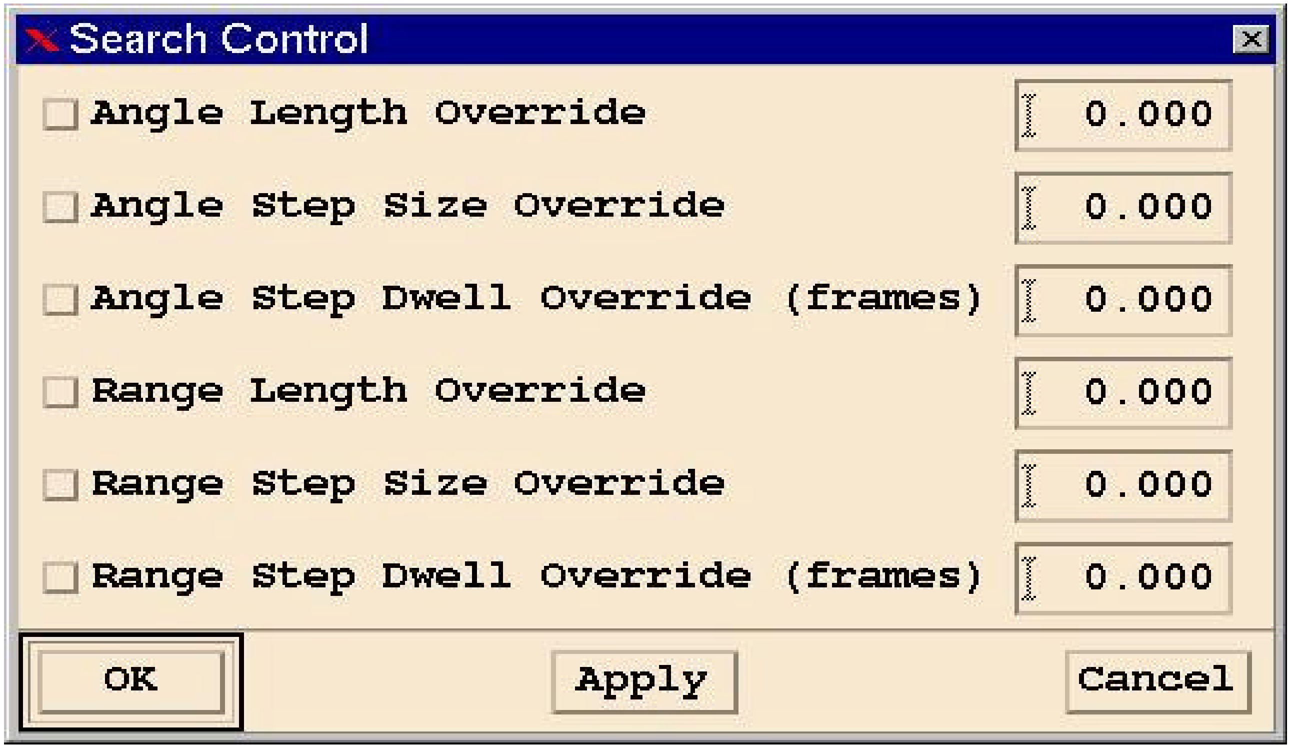


Figure 15.4.3d-Ratgui search control and override

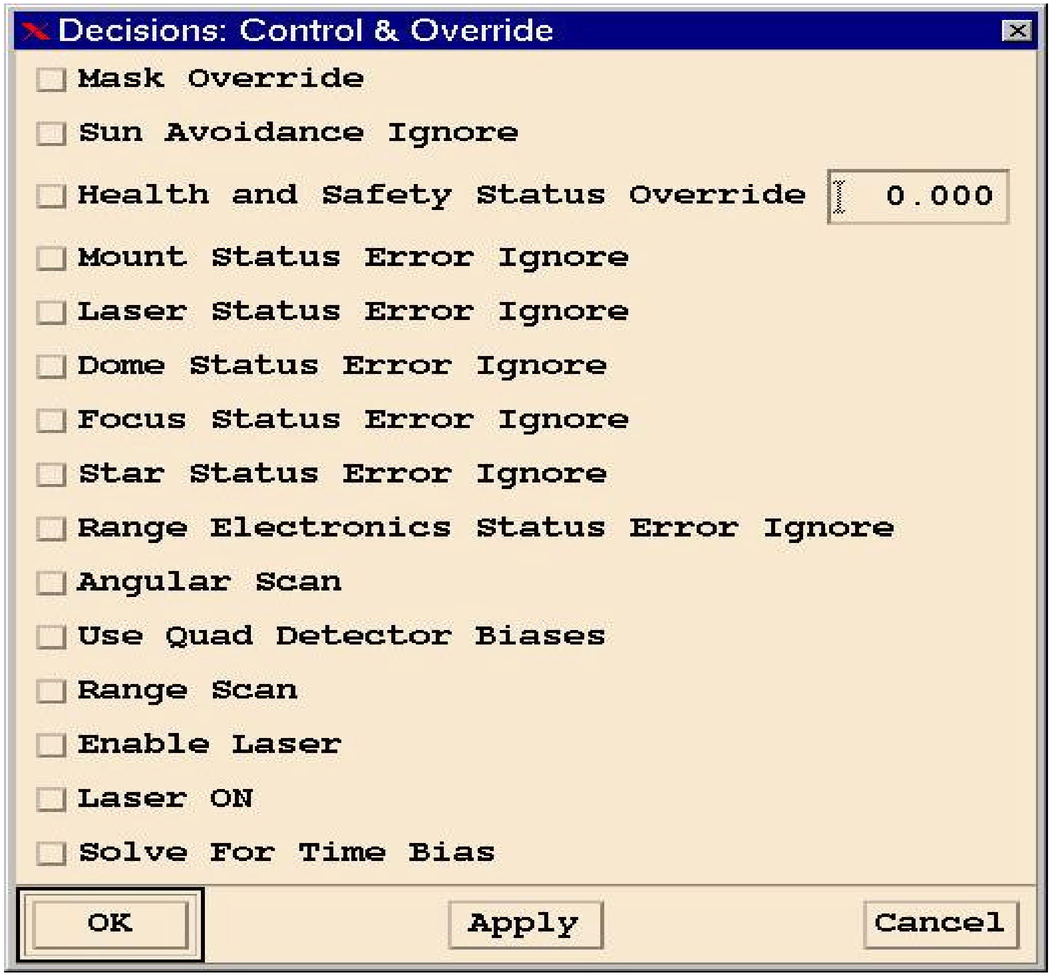


Figure 15.4.3e-Ratgui decisions control and override

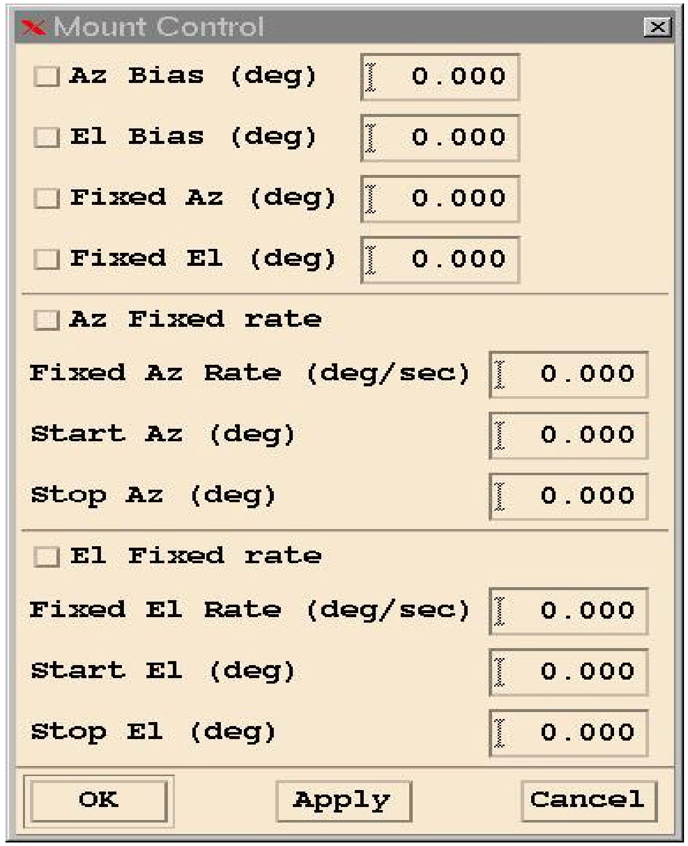


Figure 15.4.3f-Ratgui Mount control and override

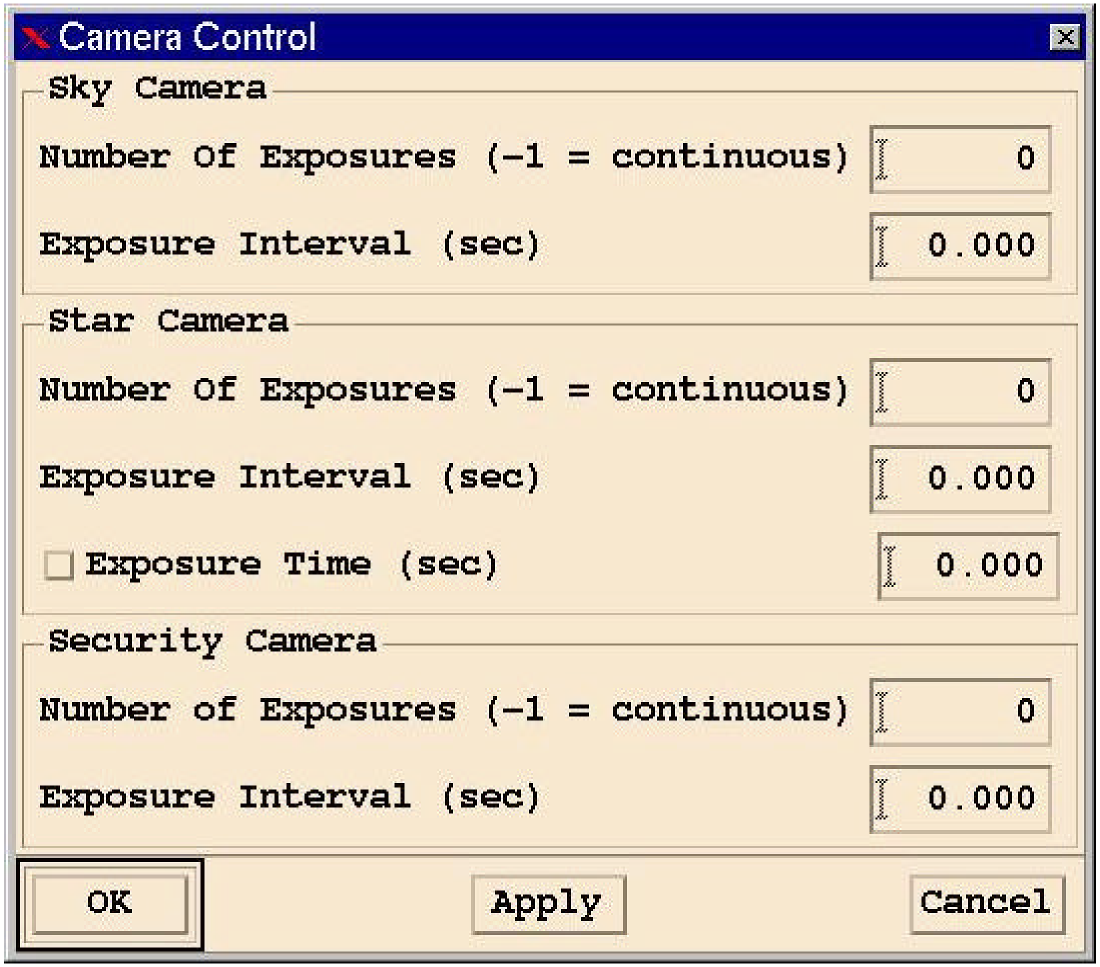


Figure 15.4.3g- Ratgui Camera Control Window

15.5) delivery schedule

15.5.1) Version 0.1: Delivered on 7/1/98

Rat ftns are run on POP

Displays:
- Range O-C
- Camera image display
- Text printouts of circular buffer and range pieces
- Timing alarms displays

Inputs:
-Turn on/off displays and text files outputs

15.5.2) Version 1.0: Will be delivered on 12/01/98

RAT runs on laptop. Diagnostics will include mount fixed pos/vel and plot errors.

Displays:
- Ranging displays
- Cameras (sky and star)
- Quadrant digitizer (4 histograms and AZ/EL biases calculated).
- Decision output/display
- Angular plots.
-Dome AZ error plot
-Alarms, to include timing error

Inputs:
- Angular search ON/OFF and apply biases manually
- Override schedule
- H&S decision override
- ACQ/TRK (apply decision vs don't) & fix RW & RD
- Use QD for bias (apply vs don't)
- Emergency STOP (mount, laser, power)
- DOME drive enabled /disabled

- AZ/EL fixed position
- Range inputs (cmd bias & fixed override, Window fixed)
- Bias to UT
- Dome AZ fixed positon & bias
-Watchdog Timer ON/Off
-Turn ON/OFF simulations
-Debug output ON/OFF

15.5.3) Version 2.0 : Will be delivered 02/15/99

Version 2.0 includes all ftns of v1.0

Displays:
- Starcal plots

Inputs:
- Starcal controls including override schedule and pause on each star
- Focus control
- Laser control (ON/OFF)
- laser PRF override

15.5.4) Version 3.0: Will be delivered 09/01/99

Version 3.0 includes all ftns of version 2.0

Displays:
- Security camera display
-Weather information display
- H&S plots
- Post-fit O-C plot

Inputs:
- Weather overrides
- H&S value overrides

16) Software development and test schedule

A) ICC CRR checkout with hardware simulator---Jul . 98

B) Star analysis checkout on the 48" telescope--Oct . 98

C) System checkout with RAT--Dec . 98

D)Mount testing at vendor---Jan . 99

E) Instrumentation moves to S2K facility---Mar . 99

F) System checkout with RAT---Apr . 99

G) Star Calibrations---Jun . 99

I) Ground calibration--Aug . 99

J) Satellite tracking attempt ---Oct . 99

Appendix

SLR 2000 Dome Control Interface

Michael Perry

AlliedSignal Technical Services Corporation

Tony Mallama, Nick Ton and Jack Cheek

Raytheon STX Corporation

October 23, 1998

This document explains the interface between the Pseudo-OPerator (POP) and the Dome Control System (DCS).

The hardware interface will be an RS232 link from the DCS to POP using the COM2 port.

The data flow between POP and DCS is described below. We will transfer data at a 1HZ rate at a minimum speed of 9600 baud with 8 data bits, 1 stop bit, and even parity. All data packets will start with a command byte and end with a carriage return/linefeed combination.

Data from POP to DCS. Minimum 3 bytes, maximum 21 bytes.

Byte1	Command byte	bit 0	1=>Close Dome, 0=>Open Dome.
		bit 1	1=>Manual Control, 0=>POP control.
		bit 2	1=>Diagnostics, 0=>Normal operation.
		bit 3	1=>Sleep, 0=>Normal operation.
		bits4-7	Unused.

The remainder of the data will vary depending on the command byte.

If the command byte is equal to a 0, this denotes normal operation we

will include the following information:

Byte2	Azimuth position	0	LSB
Byte3	“ ”	0	MSB
Byte4	Acceleration	0	LSB
Byte5	“	0	MSB
Byte6	Velocity	0	LSB
Byte7	“	0	MSB
Byte8	Azimuth position	1	LSB
Byte9	“ ”	1	MSB

Byte10	Acceleration 1		LSB
Byte11	Acceleration 1		MSB
Byte12	Velocity 1		LSB
Byte13	“		MSB
Byte14	Azimuth Position 2		LSB
Byte15	“ ”	2	MSB
Byte16	Acceleration	2	LSB
Byte17	“	2	MSB
Byte18	Velocity	2	LSB
Byte19	“	2	MSB
Byte20	Carriage Return	13	
Byte21	Linefeed	10	

If the command byte is a 1, it implies that the shutter is closed due to a normal perational event (i.e. precipitation). In this case only the command byte and the carriage return/line feed will be sent.

If the command byte is a 2, it implies that the shutter is open and that the dome will be controlled manually, i.e. via handpaddle or some other means. In this case only the command byte and the carriage return/linefeed combination will be sent.

If the command byte is 3 it implies that the shutter is closed and that the dome will be controlled manually, again only the command byte and the carriage return/linefeed combination will be sent.

If the command byte is 4, diagnostics mode, it implies that the shutter is open, and that the dome is under POP control, so the entire 21 bytes defined above will be sent.

If the command byte is 5, diagnostics mode, it implies that the shutter is closed, and that the dome is under POP control. All 21 bytes will be sent.

If the command byte is 6, diagnostics mode, it implies that the shutter is open, and that the dome is in manual control. Only 3 bytes will be sent.

If the command byte is 7, diagnostics mode, it implies that the shutter is closed and that the dome is in manual control. Only 3 bytes will be sent.

If the command byte is 8, sleep mode, it implies that POP will be down for an undetermined amount of time and that the DCS should go into a sleep mode, which implies that the shutter is closed and the dome should go to a stow position (TBD). Once in a stow position the DCS should only monitor it’s incoming RS232 to see if anything new comes along. When the DCS begins receiving data again, it implies that POP is now up and running, although we may continue to keep the system in a sleep mode. This also applies to the condition when POP should crash, if after a period of time, POP does not communicate with the DCS, it should close the shutter and move the dome to a stow position, again waiting for POP to reestablish communications.

The Data flow from DCS to POP is described below we will be transferring data at the same rates and frequencies as described above.

Byte 1	Azimuth Position	LSB		
Byte 2	“ ”	MSB		
Byte 3	Azimuth Status	bit 0	0=> Microcontroller (Dallas) is OK.	
				1=> Microcontroller (Dallas) error.
			bit 1	0=> RS232 from POP is OK.
				1=> RS232 is bad.
			bit 2	0=> Drive current is OK.
				1=> Drive current is bad.
			bit 3	0=> Drive temperature is OK
				1=> Drive temperature is bad.
			bit 4	0=> We are under POP control.
				1=> We are under manual control.
			bits 5-7 unused.	
Byte 4 Shutter Status			bit 0	0=> Microcontroller (PIC) is OK.
				1=> Microcontroller (PIC) error.
			bit 1	0=> RX from Dallas is OK.
				1=> RX form Dallas is bad.
			bit 2	0=> Drive current is OK.
				1=> Drive current is bad.
			bit 3	0=> Drive temperature is OK.
				1=> Drive temperature is bad.
			bit 4	0=> Cable strain is OK. (?)
				1=> Cable strain is bad. (?)
			bit 5	0=> Battery charger is OK.
				1=> Battery charger id bad.
			bit 6	0=> Shutter is open.
				1=> Shutter is closed.
			bit 7	0=> Battery is sufficiently charged.
				1=> Battery needs charging.
Byte 5	Carriage return	13		
Byte 6	Linefeed	10		

If we are in a diagnostics mode additional information will also be passed back to POP. We will determine this information later.